

【 信州大学審査学位論文 】

赤色及び青色LED点滅光の点滅周期と
デューティ比がユーグレナの光合成速度に
及ぼす影響

2012年9月

信州大学大学院総合工学系研究科
生物・食料科学専攻

大橋光男

目次

第1章 序論

1. 1. 本研究の背景
1. 2. 植物工場に関する既往の知見
1. 3. ユーグレナに関する既往の知見
1. 4. LED点滅光照射に関する既往の知見
1. 5. 本研究の目的

第2章 赤色LED点滅光がユーグレナの光合成速度に及ぼす影響

2. 1. 序論
2. 2. 材料及び実験方法
 2. 2. 1. 試料と培養方法
 2. 2. 2. 試料の調整方法
 2. 2. 3. 1. LED光源の仕様
 2. 2. 3. 2. LEDパネル部の構成
 2. 2. 3. 3. LEDドライブ部の構成
 2. 2. 3. 4. LED電流制御部の構成
 2. 2. 3. 5. LED点滅タイミング制御部の構成
 2. 2. 3. 6. LED点滅制御データ送信部の構成
 2. 2. 3. 7. 全体配線について
 2. 2. 3. 8. 各ソフトウェアの詳細
2. 2. 4. 光合成速度の測定方法
2. 2. 5. 光合成の光利用効率を評価する指標
2. 2. 6. 実験方法
 2. 2. 6. 1. 光-光合成曲線の測定方法
 2. 2. 6. 2. 光-光合成曲線の急峻部分の測定方法

- 2. 2. 6. 3. 光合成速度に対する点滅周期の影響の測定方法
- 2. 3. 結果
- 2. 3. 1. 光－光合成曲線
- 2. 3. 2. 「光合成PPFD効率」の最大値
- 2. 3. 3. 光合成速度に対する点滅周期の影響
- 2. 4. 考察
- 2. 5. まとめ

第3章 赤色LED点滅光の点滅周期とデューティ比がユーグレナの光合成速度に及ぼす影響

- 3. 1. 序論
- 3. 2. 材料及び実験方法
- 3. 2. 1. 試料と培養方法
- 3. 2. 2. 試料の調整方法
- 3. 2. 3. LED光源の仕様
- 3. 2. 4. 光合成速度の測定方法
- 3. 2. 5. 実験方法
- 3. 2. 5. 1. 各PPFD値における、光合成速度に対する点滅周期の影響を測定
- 3. 2. 5. 2. デューティ比の影響を測定
- 3. 3. 結果
- 3. 3. 1. 各PPFD値における、光合成速度に対する点滅周期の影響
- 3. 3. 2. デューティ比の影響
- 3. 4. 考察
- 3. 5. まとめ

第4章 青色と赤色LED点滅光の点滅周期とデューティ比がユーグレナの光合成速度に及ぼす影響

- 4. 1. 序論
- 4. 2. 材料及び実験方法
 - 4. 2. 1. 試料と培養方法
 - 4. 2. 2. 試料の調整方法
 - 4. 2. 3. LED光源の仕様、及び構成
 - 4. 2. 4. 光合成速度の測定方法
 - 4. 2. 5. 実験方法
 - 4. 2. 5. 1. 光合成速度に対する点滅周期の影響を測定
 - 4. 2. 5. 2. デューティ比の影響を測定
- 4. 3. 結果
 - 4. 3. 1. 光合成速度に対する点滅周期の影響
 - 4. 3. 2. デューティ比の影響
- 4. 4. 考察
- 4. 5. まとめ

第5章 総合考察

- 5. 1. 光-光合成曲線について
- 5. 2. 点滅光と連続光との比較について
- 5. 3. 青色と赤色LED点滅光の同時照射について
- 5. 4. まとめ
- 5. 5. 今後の課題

基礎となる論文

謝辞

引用文献

出願特許

要旨

第 1 章

序 論

1. 1. 本研究の背景

世界の総人口は産業革命以来、幾何級数的に急増しており (Donell ら, 2005a)、2050 年には 90 億人を超えると予想されている (総務省, 2011a)。また、2008年の世界総耕地面積は、2000 年と比較すると約 182 万ha 減少している (総務省, 2011b)。さらに人口増加のため、世界の「一人当たりの穀物生産量」も 1980 年代以降減少している (Donell ら, 2005b)。従って、もしもこれらが継続すれば、将来食糧不足が発生する可能性がある (Donell ら, 2005b)。

食糧不足を解決するためには、食糧増産が必要である。従来型の農業で食糧増産をするためには耕地面積の拡大が必要だが農業に適した土地の大半は既に農地化されており、残っている土地は森林や草地など限られている (Donell ら, 2005b)。仮に残っている森林を耕地化すると、貴重な二酸化炭素吸収源である森林の破壊に繋がる問題があるため、耕地面積の拡大には限界がある (Donell ら, 2005c)。従って、耕地面積を増加せずに、食糧生産量を増加させること、すなわち単位面積当たりの収量を飛躍的に高めることが必要である。しかし、従来農法では品種改良や化学肥料、農薬などで収量を増加させた経緯があるものの、これ以上の飛躍的な収量増加は困難な状況である。これは、長期的な視点から考えると、全世界の人類にとって極めて重大な課題である。

上述は食糧の「量」の問題であるが、「質」の問題も懸念されている。具体的には、日本国内で輸入農産物の残留農薬が問題になる中、2006 年から残留農薬ポジティブリスト制が施行されている (高辻, 2007a)。従って、「食の安全・安心」という「質」の問題への社会的な要求も高まっている。しかし、従来型の農業では「食の安心・安全」を図るために、無農薬で作物を栽培すると病気や害虫による被害が発生したり雑草が繁茂するため、収量が減少するという問題がある。

仮に、農薬が出現する以前の様に、無農薬農法で人手に頼り雑草や害虫を駆除したとしても、全世界の農地で実施するためには、膨大な作業工数が発生するとともに、無農薬農法は基本的に収量増加に繋がらないため、食糧問題解決の現実的な手段とは言え

ない。この様に、地球上で限定された耕地面積などの制約条件下で、人口爆発する人類の食糧需要の「量」と「質」を将来に渡り満足することは非常に困難な状況である。

ここで農業の歴史を振り返ると、18世紀の英国で始まった「輪作」や「圃い込み」を皮切りに、その後「三圃制」や「輪栽式農業」等のいわゆる農業革命により生産量を段階的に向上させてきた歴史がある。この歴史の流れの様に、今後人類の食糧需要を満たすためには、従来農法とは異なる新たな農業革命が必要である。

1. 2. 植物工場に関する既往の知見

近年、従来農法と異なる農法として「植物工場」と呼ばれる新農法が実用化されている。植物工場とは、「野菜や苗を中心とした作物を施設内で光、温湿度、二酸化炭素濃度、培養液などの環境条件を人工的に制御し、季節や場所に関係なく自動的に連続生産するシステム」と定義されている（高辻，2006）。

植物工場では、光量や日長、温度、湿度、二酸化炭素濃度などの環境条件を植物栽培に最適な値に制御可能なため、たとえば、サラダナを用いた実験において露地栽培と比較して約5～6倍の成長速度を実現した（高辻，2006）。さらに、植物工場では閉鎖環境にすることで病害虫との隔離が可能である。このため、完全無農薬で安全・安心な作物を場所や天候に左右されずに安定的に周年生産することが可能である（高辻，2006）。

具体的には、植物工場を都会のビルや地下空間にも設置可能なため、消費地間近での生産が可能である。さらに、各栽培棚と照明を交互に積み重ねて3次的に配置した多段式栽培により土地の利用効率を飛躍的に向上させることも可能である（高辻，2006）。植物工場の収量増大に着眼した場合、栽培棚の多段式栽培の他に、どの様な品種の植物を栽培するのか、という栽培品種も収量増大に大きく影響する要素である。例えば露地栽培において多収品種を栽培すれば収量が増加することと同様である。従って、植物工場での収量をさらに増量させるためには、多収品種を選択することも重要である。また、藁などの非可食部は、食糧に出来ないため植物バイオマスの利用効率

の視点から考えると、非可食部の比率は極力少なく可食部の比率が多い方が収量増大に繋がる。

さらに、エネルギーの視点から考えると、植物工場は太陽光に代わる強力な照明を必要とするため、大きな電力エネルギーを消費するという課題がある（高辻，2006）。地球上では、耕地面積と同様に総電力エネルギーも有限なリソースであり、2011年の東日本大震災以来、さらなる節電が叫ばれている。

従って、収量の他に、照明用電力エネルギーの視点から栽培時のエネルギー効率を考慮することも重要である。基本的に植物が必要とするエネルギーは、光合成に必要な「光エネルギー」である。従って、光エネルギーの利用効率が高い栽培品種、すなわち、「光合成の光利用効率」が高い品種を選択することが重要である。「光合成の光利用効率」が高いということは、単位面積当たりに照射された光を効率的に光合成へ利用出来ることであるため、収量増大にも繋がる。

1. 3. ユーグレナに関する既往の知見

近年、「光合成の光利用効率」が非常に高い植物として、ユーグレナ (*Euglena*) という藻類が注目されており、新たな食資源としての活用が提案されている（中野ら，1998）。ユーグレナという学名は、ラテン語のeu（美しい）とglena（瞳）に由来している。和名では「ミドリムシ」と呼ばれ、学校教育にて顕微鏡観察の試料に利用される身近な生物である。

ユーグレナには、葉緑体があり光合成を行う。さらに、ユーグレナには、鞭毛があり、この鞭毛を動かして液中を移動するという動物的な能力もある。ユーグレナ属および近縁の属種は約120種あり、その多くは田園や湖沼の様に流れの少ない淡水域に生息している。形状は、長径50～100 μm で短径10～50 μm の紡錘形であり、鞭毛を回転させて光の方向へ移動する性質がある（中野ら，2012）。ユーグレナの細胞外膜（ペクリル）は、タンパク質を主成分としており、表面に螺旋状の条溝を多数有する。この螺旋状の条溝があるため、紡錘形から球形へと双方向に変形する

「Euglenoid運動」と呼ばれる形状変化をする特徴もある（北岡ら, 1989b）。

ユーグレナにおける「光合成の光利用効率」を穀物と比較すると、穀物の約 50 倍である。この約 50 倍という圧倒的に高い生産能力は、驚愕に値する。具体的には、ユーグレナに適した光量や温度、培養液などの環境で栽培（培養）することにより、1 ha 当たり、年間約 250 t の収穫物（乾燥藻体）を生産可能である。また、米国における 1 ha 当たりの穀物生産量の年間平均値は約 4.6 t である。従って、1 ha 当たりのユーグレナの年間収量と米国における穀物の年間収量の比率を算出すると、54.3 であり、約 50 倍である（中野ら, 1998; 宮武ら, 1999; 松本ら, 2006）。

単位面積当たりの生産量（収量）が 50 倍になるということ、視点を変えて逆算すると、「同じ生産量を得るために必要な土地の面積が 1 / 50 で済む」ということである。例えば、将来人口爆発のため現在の 2 倍の食糧を生産する場合、2 倍 / 50 = 1 / 25 となるため、必要な耕地面積は、従来の 1 / 25 = 4 % で済む計算になる。すなわち、耕地に余剰面積が発生することになり、例えばこの余剰面積部分に植林を行い森林に戻すことで、二酸化炭素濃度増加などの環境問題を緩和可能と期待出来る。

さらに、ユーグレナは高濃度の二酸化炭素を効率良く吸収する能力がある。このため、二酸化炭素を大量に排出する火力発電所とユーグレナ培養システムを組み合わせ、発電所が排出する二酸化炭素をユーグレナに吸収させれば、発電所の煙突から大気中に放出される二酸化炭素を直接的に削減可能である。実際に（株）ユーグレナが 2009 年に火力発電所からの排出ガスを用いて、ユーグレナを培養する実証試験を実施した。その結果、火力発電所の排出ガスを通気しての培養に成功した（弟子丸, 2012）。さらに、ユーグレナの活用は下水処理や航空機のバイオ燃料にまで広がりつつある（出雲 充, 2012; 弟子丸, 2012）。

また、ユーグレナは確認されているだけでもビタミン、ミネラル、食物繊維、機能性不飽和脂肪酸といった 59 種類の栄養素が含まれており、非常に高い栄養価がある（中野ら, 1998; 松本ら, 2007, 2009; Sugiyama ら, 2009）。さらに、ユーグレナには、人間が消化しにくい細胞壁がないため豊富な栄養素が消化され易いと考えられ

ている（中野ら，2012）。

特にユーグレナに特異的に含まれているパラミロンは、腸内の余分な脂質等を吸着する食物繊維様の働きが期待されている。パラミロンは、グルコースが約 700 個程度重合した β -1,3-グルカンの結晶構造をしており、ユーグレナの細胞内では長径約 1 ~ 5 μ m の紡錘形の顆粒構造である。マウスを用いた実験ではパラミロン投与群において、アレルギー性皮膚炎を緩和する機能があると報告された（中野ら，2012）。さらに、脳卒中易発症性自然発症高血圧ラット（SHR-SP）の寿命延長効果も報告された（北岡，1989a）。この様に高い機能性を有した素材であることから、機能性食品としての利用も進んでいる。

この様な特長があるため、ユーグレナを培養することで高栄養価な食資源を効率的に供給することが可能になり食糧不足を解決出来る可能性がある（中野ら，1998；宮武，1999）。ユーグレナは培養液中での高密度培養が可能であり、培養液の温度や組成、光条件等を適正化すれば閉鎖系で工業的に培養可能（宮武，1999）であるため、植物工場に適した生物である。

しかし、本格的に大規模な培養システムを実現するためには、大量培養を可能にする新たな技術開発が必要である（藤田・青柳，2012）。具体的には、大型培養槽や効率的な照明装置などの技術開発が必要である。

1. 4. LED点滅光照射に関する既往の知見

植物用の効率的な照明方法として、LED点滅光を照射することがサラダナを用いて検討されており、連続光を照射した場合と比較して光合成速度や生長率が増加したと報告された（森ら，2002a, 2002b）。LED点滅光とは、LED（Light Emitting Diode, 発光ダイオード）を用いて点灯と消灯を規則的に繰り返して発光させた際に得られる点滅した光のことである。

LEDは半導体の一種であり、P型半導体とN型半導体を接合した「PN接合」を基本構造としている。LEDの発光原理は、白熱電球や蛍光灯とは異なり、PN接合部に順方向電

圧を印加することで、各半導体領域にそれぞれの少数キャリアを注入し多数キャリアと再結合する際に、バンドギャップに相当する波長の光子が発生するという発光原理である（西澤ら, 1999）。少数キャリアを注入することとは則ち、P型半導体領域には電子を注入し、N型半導体領域には正孔を注入するということである。この様にLEDは、半導体中での電子と正孔の再結合により発光するため、白熱電球や蛍光灯とは異なり、応答速度は発光再結合寿命によって決まるため、通常 $0.1 \mu\text{s}$ 以下の高速な点滅動作が可能になる（西澤ら, 1999）。

既報では、サラダナにLED点滅光を照射し、点滅周期を $2 \mu\text{s} \sim 10 \text{ms}$ まで段階的に変化させた場合と、連続光を照射した場合の成長率が比較された。その結果、点滅周期を $400 \mu\text{s}$ にした場合に連続光よりも成長率が 23% 増大したと報告された。さらに、点滅周期と明期との比率である点滅デューティ比を $10 \sim 100\%$ まで段階的に変化させた場合の成長率が比較された。

その結果、点滅デューティ比を 33% にした場合に連続光よりも成長率が 25% 増大したと報告された（森ら, 2002a, 2002b）。さらに、点滅周期 $400 \mu\text{s}$ で光合成速度が増大した要因として「光化学系Ⅱの反応中心クロロフィル (P680) の還元時間が $200 \mu\text{s}$ で律速条件となり、点滅周期 $400 \mu\text{s}$ とした場合のパルス幅 ($200 \mu\text{s}$) と一致すること」と述べている（森ら, 2002a, 2002b）。すなわち、サラダナにおいて、葉緑体の光化学系Ⅱに適合したLED点滅光を照射することにより、連続光と比較して、光合成速度や生長率を増加させることが出来る。サラダナと同様にユーグレナにも葉緑体があり光合成を行っていることが知られている（横田, 1989）。従って、ユーグレナに対してもLED点滅光を照射することにより、サラダナの例と同様に、光合成速度や生長率を増加させることが期待出来る。この様に、LED点滅光は植物栽培用として非常に有効な照明方法である。

補足だが、本来ユーグレナに近縁な藻類などを用いた報告例を参考にすべきであるがその様な報告例はなかった。

1. 5. 本研究の目的

将来ユーグレナを効率良く大量に培養出来れば、上述した様に食糧問題や二酸化炭素問題などを解決出来るため、全人類に対して甚大な貢献が期待できる。しかし、2005年に太陽光を利用した屋外（開放池）での培養が成功したものの、将来の需要増加を考えるとさらなる大規模化と効率化が必要である（藤田・青柳, 2012）。だが、さらなる大規模化や効率化を実現するためには、培養液の温度や各栄養成分の濃度、二酸化炭素濃度などの培養環境制御、生産効率向上、雑菌増殖の抑制、照明技術など様々な技術課題が山積しており、これらの技術課題を解決する必要がある（藤田・青柳, 2012）。

ユーグレナの培養方法には大きく分けて、「開放系」と「閉鎖系」という培養方式がある。「開放系」とは、外界に対して開いた方式であり、太陽光を光源とした循環池や水路池などの方式がある。「閉鎖系」とは、外界から隔離した閉鎖環境で培養する方式であり、チューブ型やパネル型などの形状がある（藤田・青柳, 2012）。両者には長所と短所があり、「閉鎖系」は雑菌の混入がなく最適な環境を維持出来るが、「開放系」と比較して設備コストが高額になる場合が多い。「開放系」では設備コストは安価だが雑菌混入や培養効率が著しく低いことや天候等の外部環境変化の影響を受けバイオマス収量が不安定になる等の問題がある（藤田・青柳, 2012）。

従って、火力発電所や工場の煙突から排出される高濃度な二酸化炭素を含んだガスを効率的にユーグレナに固定化させるためには、大規模な「閉鎖系」の培養システムが必要である（藤田・青柳, 2012）。仮に、開放系でユーグレナに二酸化炭素を固定化させた場合には、火力発電所や工場から排出される高濃度な二酸化炭素を含んだガスを大気中に漏洩する恐れがある。しかし、現在これを可能とする大規模な閉鎖系の培養システムは実現していない。これを実現するためには、培養液の温度や各栄養成分の濃度、二酸化炭素濃度などの培養環境制御、照明技術などの技術課題を解決する必要がある。

これらに技術課題の中から、筆者らは、特に「照明技術」に着眼した。なぜならば「照明技術」が提供する光環境は、ユーグレナにとって全生産活動の源動力とも言え

る根源的な「光合成反応」に必須な要素だからである。特に「光合成の光利用効率」を向上させることは、最終的なバイオマス収量に大きく影響する。

従って筆者らは、ユーグレナの「光合成の光利用効率」を向上させる光環境を実現可能な「照明技術」を明らかにすることを本研究の目的とした。「照明技術」とは、則ち光量や光の波長、連続光、点滅光、点滅周期、点滅デューティ比、点滅明期間の光量、複数の光波長の組合せ比率などの光質条件に関する一連の技術のことである。既報にてLED連続光のみをユーグレナに照射した報告 (Aiga ら, 1997) があるが、点滅光を用いて点滅周期や点滅デューティ比の影響を報告した例はない。

従って、ユーグレナに対して、LED点滅光を照射することで、点滅周期や点滅デューティ比、光量などの光質条件がユーグレナの光合成速度へ及ぼす影響を明確化することが必要である。具体的には、まず第一にユーグレナに対してLED点滅光を照射しながら光量を広範囲に変化させることで「光-光合成曲線」の概要を調べることが必要である。つぎに、得られた「光-光合成曲線」の急峻部分（光合成効率が低い部分）を詳細に調べることで、「光合成効率が最も高くなる光量条件」を明確化する必要がある。さらに、得られた「光合成効率が最も高くなる光量条件」に基づき、点滅光の点滅周期を変化させながら「光合成速度に対する点滅周期の影響」を測定することで「光合成効率が最も高くなる点滅周期条件」を明確化する必要がある。

つぎに、得られた「光合成効率が最も高くなる点滅周期条件」に基づき、点滅光の点滅デューティ比を変化させながら「光合成速度に対する点滅デューティ比の影響」を測定することで「光合成効率が最も高くなる点滅デューティ比条件」を明確化する必要がある。さらに、光量や点滅周期、デューティ比の各条件因子が相互依存性を有する場合があるため、得られた「光合成効率が最も高くなる点滅条件等」に基づき、再度、光量を変化させながら各点滅条件の「光-光合成曲線」を調査する必要がある。ここまでの調査では、光源として、光合成に最も効率的な波長である赤色光 (高辻, 2007b) のみを使用した。しかし、エマーソン効果 (Emerson・Rabinowitch, 1960) の様に複数の光波長 (青色と赤色など) を同時に照射することで光合成効率がさらに向上したと

が報告されている（松本ら，2006，2009）。

従ってつぎに、ユーグレナに対して青色と赤色のLED点滅光を同時に照射しながら、光量や点滅周期などを変化させることで「光合成効率が最も高くなる青赤混合照射条件」を明確化する必要がある。従って、筆者らは、上述の順にユーグレナに対してLED点滅光を照射することで「光合成効率が最も高くなる光質条件」を検索した。

これにより、ユーグレナに最も適した効率的な光質条件を明らかにすることを目指した。ユーグレナに最も適した効率的な光質条件を明らかにすることが出来れば、将来ユーグレナを大規模に大量培養するシステムである、「ユーグレナ用植物工場」の実現へ向けて大きく前進することになる。このことは、深刻な環境問題に直面した全人類にとって大きな一歩になると期待する。

第2章

赤色LED点滅光がユーグレナの
光合成速度に及ぼす影響

2. 1. 序論

「第1章」で述べた様に、ユーグレナを活用することで全人類の課題である食糧問題や二酸化炭素問題などの解決が見込まれ、甚大な貢献が期待出来る。 これを実現するためには、ユーグレナを閉鎖系で連続的に大量培養することが必要である。 しかし、光質条件など多くの技術課題があるため、ユーグレナを閉鎖系で連続的に大量培養することは、現在実現出来ていない。 従って、効率的な培養を可能とする光質条件などの技術課題を解決することが必要である。 ここで植物の効率的な栽培事例をあげると、LED点滅光条件下でサラダナを栽培した場合には、連続光を照射した場合と比較して光合成速度が増加したと報告された（森ら、2002a, 2002b）。

そこで筆者らはこの「LED点滅光」をユーグレナに応用すればさらなる効率的な培養が可能になるのではないかと着想した。 これを検証するため、ユーグレナにおけるLED点滅光での光合成効率、すなわちLED点滅光での「光-光合成曲線」を測定した。 測定に用いたLEDの種類は赤色LED（波長約 660 nm）を使用した。 赤色LEDを使用した理由は、光合成を行う葉緑体のクロロフィルaがこの波長の光を効率良く吸収するため、さらなる光合成の効率化が期待出来るからである（高辻、2007b）。 つぎに、赤色LED連続光での「光-光合成曲線」を測定した。 目的は、赤色LED点滅光を照射した場合と連続光を照射した場合の光合成速度を比較することで、より高効率な光合成条件を検証するためである。

また、ユーグレナに照射する光量は光合成効率に大きく影響する要素である。 例えば、光量が光補償点よりも弱ければ育たないし、逆に光量が強過ぎれば光阻害や光飽和が発生し光合成効率が低下する。 従って、ユーグレナに対して最適な光量を照射することは、光合成効率を向上させる上で重要である。 実際の商業生産を考慮すると1%でも高い効率や生産性が求められるため、極力、高精度に光量の最適条件を求める必要がある。 しかし、過去にユーグレナに関して高精度に光量の最適条件を報告した例はない。

従って筆者らは、ユーグレナに照射する光量の最適値をより高精度に調べた。 す

なわち、単位光量当たりの光合成速度が最も高くなる最適な光量条件を調べた。具体的には、最適な光量条件を正確に求めるため、設定条件であるPPFD値の設定値の間隔を細かく設定して測定した。上述の調査内容の詳細を以下に報告する。

2. 2. 材料及び実験方法

2. 2. 1. 試料と培養方法

測定用試料には、ユーグレナ (*Euglena gracilis* Z 株) を培養し用いた。具体的には培養槽として 5000 ml のセパラフラスコ (No30050-07, Vidrex Co., Ltd) を用い、これに 3000 ml のCramer-Myers培地 (CM培地) (Cramer・Myers, 1952) (Table. 2-1) を入れ、セパブルカバー (No3280-04, Vidrex Co., Ltd) で蓋をしてオートクレーブにて滅菌処理後、クリーンベンチにて無菌的にユーグレナを注入した。

さらに、エアーポンプ (P-15A, 株式会社 安永) とメンブレンフィルター (SLFG05010, Millipore Co., Ltd) で濾過した空気を培養槽へバブリングし、蛍光灯にて光を照射し培養を行った。培養中の環境条件は気温 25 °C、培養液温度 25 °C、光合成有効光子束密度 (以降、PPFDと記載する) $50 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ とした。PPFD値の測定は、セパラフラスコの内側に光量子センサーを配置し、光源である蛍光灯と光量子センサーとの距離を約 0.15 m とし、PPFD値の実測値が $50 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ になる様に距離を微調整した。

PPFD値の測定には、光量子センサ (Quantum Sensor LI-190, LI-COR Ltd) 及び、データロガー (LI-1400, LI-COR Ltd) を使用した。また、培養中は定期的に培養液を抜き取り、新鮮な培地を追加した。これにより培養液の細胞濃度を約 $1.0 \times 10^5 \text{ cell}\cdot\text{ml}^{-1}$ 以上に保持した。

2. 2. 2. 試料の調整方法

培養槽から無菌的に 1 ml の培養液をサンプリングし、これにCM培地 1 ml と NaHCO_3

溶液 (0.5 M) 80 μ l を入れた溶液を作成した。この溶液をクラーク電極タイプの液相酸素濃度測定器 (OXY-1, Hansatech Instruments Ltd) の測定チャンバーへ入れた。測定チャンバーの形状は円筒形で、直径 10 mm \times 高さ約 32 mm であった。サンプリング以降の作業はクリーンベンチ内にて、メンブレンフィルター (DISMIC-25CS, 東洋濾紙株式会社) や消毒薬 (アセサイド 6 % 消毒液, サラヤ株式会社) を用いて無菌状態で行った。

その後チャンバーを遮光し暗呼吸を促し、チャンバー内の溶液中酸素濃度が初期値 (約 100 $\text{nmolO}_2\cdot\text{ml}^{-1}$) に安定化するまで待ち光合成速度の測定を開始した。

2. 2. 3. 1. LED光源の仕様

実験に用いる光源として、下記仕様のLED光源ユニットを製作した。

- ① PPF値 : 0 から約 1000 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ まで任意に設定可能。
- ② 照射光の中心波長 (λ) : 660 nm (赤) (分光特性: Fig. 2-1)
- ③ 点滅周期 (t_2) : 100 ns \sim 48 h (範囲内を 10 ns 単位で設定可能) (Fig. 2-2)
- ④ 1 周期当たりの点灯時間 (t_1) : 0 \sim t_2 (周期内を 10 ns 単位で設定可能)

t_1 を変化させることで「点滅光のデューティ比」を 0 \sim 1 の範囲で任意に設定可能。点滅光のデューティ比とは、「 t_1 」を「 t_2 」で除した値であり、式で記載すると以下になる。

$$\text{点滅光のデューティ比} = t_1 (\text{s}) / t_2 (\text{s}) \quad (\text{式2-1})$$

補足だが、点滅点灯での光合成速度を正確に測定するためには、「10 ns 単位」で精密に点滅タイミングを制御可能であることと、点灯時の光量を正確に調整可能な光源ユニットが必要である。しかし、この条件を満足する市販品の光源ユニットは、なかった。従って、新規に光源ユニットを電子回路から設計し製作した。

2. 2. 3. 2. LEDパネル部の構成

LEDパネル部は、LED素子（GL3UR8, SHARP Co., Ltd）をプリント基板に 600 個実装し用いた。プリント基板の材質は、ガラスエポキシ製とし、表面と裏面に銅箔パターンを配置し、全板厚を 1.6 mm とした。LED素子を効率良く冷却する目的で、裏面側に熱導電性と絶縁性に優れた放熱用シリコンシートを貼り、さらにアルミニウム製の大型放熱板を 2 個取り付けた。各大型放熱板には、強制空冷用ファンを取り付けた。これらの冷却機構によりLEDを強制的に効率良く冷却した（Fig. 2-3 a, b）。プリント基板の中央部に 120 ピンのカードエッジコネクタを配置した。このコネクタに「LEDドライブ基板」（後述）を接続することで、各LEDに電流を供給するという構造である。

プリント基板上での各LEDの配線は、まず 1 列目に並んだ 10 個のLEDを直列に接続し、両端のプラス側端子とマイナス側端子をカードエッジコネクタの異なる端子へ接続した。同様に残りの各LEDも列毎にカードエッジコネクタの異なる端子へ接続し、全 60 列の直列回路をカードエッジコネクタへ接続した。この様に、10 個のLEDを直列接続した列を 60 列作り各端子をカードエッジコネクタへ接続することで、全 600 個のLEDに電流を供給する回路を製作した（回路図：Fig. 2-4）。

2. 2. 3. 3. LEDドライブ部の構成

LEDドライブ部は、LEDドライブ基板で構成されている。LEDドライブ基板の材質は、ガラスエポキシ製とし、表面と裏面、及び内層に 2 層の計 4 層の銅箔パターン層を配置し、全板厚を 1.6 mm とした。出力段のトランジスタを冷却するため大型の放熱板を配置し、さらに放熱板に強制空冷ファンを取り付けて強制的に冷却した（Fig. 2-5）。LEDドライブ基板の電子回路構成は、基本的にLEDドライブ回路が 6 チャンネル分（CH0 ~ 5）ある（回路図：Fig. 2-6）。

従って、各LEDを 6 チャンネルのグループに分割し、各チャンネル毎に独立してLEDをドライブ可能にした。こうすることで、例えば異なる波長のLEDを 6 種類使用する

場合に、LEDの種類毎に異なるチャンネルへ割り当てることで、各種類のLEDの点灯状態を独立に制御することが出来る。従って、様々な光質条件の実験が可能になる。補足だが、全チャンネルに同じ種類のLEDを割り当て、全チャンネルで同じ制御をすることも出来る。

回路図のCN4 は、ドライブ信号を出力するコネクタであり、LEDパネル部のCN1 へ接続する。CN3 は、各チャンネル毎の点灯消灯を制御する差動信号 (CH0 ~ 5) を入力するコネクタである。CN4 は、各チャンネル毎の点灯時の電流値 (光量) を制御する信号入力 (V-ON0 ~ 5) と、電流のレンジを切り替える信号 (I-x00mA CH0 ~ 5) を入力するコネクタである。CN2 は、ドライブ基板動作の電源を入力するコネクタである。CN6 は、LEDドライブ用の電源を入力するコネクタである。

回路動作の概要は、CN3 の差動信号入力 (CH0 ~ CH5) がON状態ならば該当するチャンネルのLEDを点灯し、OFF状態ならば消灯する。点灯時の電流値 (光量) は、CN4 の点灯時の電流値制御信号 (V-ON0 ~ 5) と電流レンジ切換信号 (I-x00mA CH0 ~ 5) の状態によって決まる。この電流レンジ切換信号と電流値制御信号を巧みに用いた制御方式により高精度な電流制御を実現した。

さらに、出力段にて 2 個のトランジスタをカスコード接続回路構成にしたことで、高速な点滅制御を実現した。さらに、出力段のトランジスタに対して並列に抵抗器を接続することで、本トランジスタへの印加電圧を軽減させた。

2. 2. 3. 4. LED電流制御部の構成

LED電流制御部は、LED電流制御基板で構成されている。基板の材質は、ガラスエポキシ製とし、表面と裏面の計 2 層の銅箔パターン層を配置し、全板厚を 1.6 mm とした (Fig. 2-7)。基板の電子回路構成の概要は、制御用中央演算IC (U4: PIC-18F452) とLED点灯時の電流値制御用のアナログ信号を生成するD/Aコンバータ (U3: MB88347)、LEDの電流レンジ切換信号を生成するI/O Expander (U8~9: PCF8574)、各種設定情報を記憶する不揮発性メモリー (U6: AT24C1024)、設定入力用スイッチ (SW2: DIP-SW)

である（回路図：Fig. 2-8）。

CN1 は、各チャンネル毎のLED点灯時の電流値制御用のアナログ信号（V-ON0 ～ 5）と、電流のレンジを切り替える信号（I-x00mA CHO ～ 5）を出力するコネクタである。回路動作の概要は、設定入力用スイッチ（SW2）を操作して各チャンネルの点灯時の電流値や電流レンジを設定すると、中央演算IC（U4）が設定情報に応じて、D/Aコンバータ（U3）とI/O Expander（U8 ～ 9）を制御する。その結果、各チャンネル毎のLED点灯時の電流値制御用のアナログ信号（V-ON0 ～ 5）と、電流のレンジを切り替える信号（I-x00mA CHO ～ 5）を生成し出力する。

2. 2. 3. 5. LED点滅タイミング制御部の構成

LEDを点灯及び消灯する点滅タイミング制御部は、FPGA（Field-Programmable Gate Array）基板で構成されている。基板の材質は、ガラスエポキシ製とし、表面と裏面、及び内層に 2 層の計 4 層の銅箔パターン層を配置し、全板厚を 1.6 mm とした。

（Fig. 2-9）。基板の電子回路構成の概要は、FPGA（U1：EPM2210F256）と、差動出力信号生成用の差動ライントランシーバIC（U12～29：ADM1485）、基本動作クロック生成用の水晶発信器（OSC2：SG8002CA-100M）である（回路図：Fig. 2-10）。

CN8 ～ 10 は、各チャンネル毎のLED点滅タイミング制御用の差動信号を出力するコネクタであり、3 個のコネクタから 3 系統のLEDドライブ基板へ信号を供給可能である。このため、必要に応じて複数のLEDユニットを点灯させることが出来る。従って、LEDユニット数を増加することで、今後の実験規模拡大やさらなる光量増加へも対応可能である。CN2は、CPU基板（後述）からU1（FPGA）へLEDの点滅制御に必要な設定データを書き込むためのコネクタであり、8 ビット幅のデータバスを用いている。

回路動作の概要は、CPU基板からCN2 を経由してU1（FPGA）へLEDの点滅制御に必要な設定情報を書き込む。すると、U1（FPGA）内部にてLEDの点滅タイミング制御用の信号を生成する。生成された点滅タイミング制御信号は、差動ライントランシーバIC

へ送られ、差動出力信号に変換される。差動出力信号に変換されたLED点滅タイミング制御信号は、出力用コネクタ（CN8 ～ 10）からLEDドライブ基板へ出力される。

2. 2. 3. 6. LED点滅制御データ送信部の構成

LED点滅制御用の設定データをFPGA基板へ送信する、LED点滅制御データ送信部は、CPU基板（#1）で構成されている。CPU基板の材質は、ガラスエポキシ製であり、表面と裏面の計 2 層の銅箔パターン層を配置し、全板厚を 1.6 mm とした（Fig. 2-11）。基板の電子回路構成の概要は、中央演算IC（U1～2：PIC-18F452）と拡張I/O用コネクタのCN14、拡張I/O制御用IC（U7：PALCE22V10）、I2Cバス拡張コネクタのCN10、上位コンピュータとRS232C仕様で接続するコネクタのCN7 である（回路図：Fig. 2-12）。CN14 は、FPGA基板へLEDの点滅制御に必要な設定データを書き込むためのコネクタであり、8 ビット幅のデータバスを用いている。

回路動作の概要は、中央演算IC（U2）がCN7 を経由してRS-232Cの通信プロトコルにて上位コンピュータ（パーソナルコンピュータ等）と通信し、必要な設定データを受信する。U2 は、受信した設定データをI2Cバス上のS-RAM（後述のCPU基板（#2）のU9 ～ 10：PCF8570P）へ書き込み保存する。この様にU2 は、主に上位コンピュータとの通信処理を担当している。つぎに、中央演算IC（U1）は、I2Cバス上のS-RAM（CPU基板（#2）のU9 ～ 10：PCF8570P）の内部データを読み込みFPGA基板用のデータ形式に加工する。加工されたデータは、CN14（拡張I/O用コネクタ）を経由して、FPGA基板へ送信される。

2. 2. 3. 7. 全体配線について

上述した、LEDパネル部やLEDドライブ部、LED電流制御部、LED点滅タイミング制御部、LED点滅制御データ送信部の各プリント基板や電源ユニットなどを総合的に接続した（回路図：Fig. 2-13 a～c）。図中の「CPU基板（#2）」は、U9 ～ 10 にS-RAM（PCF8570P）を実装し、これをI2Cバス経由にて利用しているのみである（Fig. 2-13 b）。

また、LED点灯用の電源電圧は、端子台（端子台 2 ～ 6）にて 36 v か 60 v のどちらかの電圧を選択可能にした（ Fig. 2-13 c）。 こうすることで、もしも青色や紫外線LEDなどの様に順方向電圧（Vf）が高い電圧のLED素子を使用する場合や、赤色LEDの様にVfが低電圧なLED素子を使用する場合にも対応可能とした。

LED点灯用の電源電圧は、各LEDのチャンネル毎に個別に設定可能とした。 従って、各LEDのチャンネル毎に、赤色や青色の様に異なるVf電圧のLED素子を使用可能である。 補足だが、LED点灯用の電源電圧を高い電圧に設定した状態で、赤色LEDなどの低いVf電圧のLED素子を点灯させることも可能だが、電流設定値を大きくした場合に、LEDドライバ基板の出力段のトランジスタ（Q1, Q2, Q5, Q6など）の発熱量が増大するため、これらのトランジスタが破壊する恐れがある。 従って、トランジスタの破壊防止のため、LED点灯用の電源電圧を選択可能とした。

2. 2. 3. 8. 各ソフトウェアの詳細

各プリント基板に搭載したプログラマブルなICの動作を決定するソフトウェアの中で、特にLEDの点滅制御に深く関係する部分に関して説明する。 LED電流制御基板のU4（PIC18F452）の主プログラムの処理概要は、まず設定入力用スイッチ（SW2）の状態を読み取り、その状態に応じて各LEDチャンネルの点灯時の電流値や電流レンジを設定するという処理である。 さらに、LED電流制御基板のU4（PIC18F452）の主プログラムに付随した「副プログラムモジュール」があり、主プログラム処理を補助している。

CPU基板（#1）のU1（PIC18F452）の主プログラムの処理概要は、まずI2Cバス上のS-RAM（CPU基板（#2）のU9～10）の内部データを読み込みFPGA基板用にデータ形式を加工する。 つぎに加工したデータをFPGA基板へ送信するという処理である。

FPGA基板のU1（EPM2210F256）のプログラムの処理概要は、まずCPU基板から送信されたLEDの点滅制御に必要な設定データをU1の内部レジスタへ読み込む。 読み込まれた設定データに従い、LED点滅タイミング制御用の信号を生成するという処理である。 具体的には、点滅周期や明期、暗期など設定データに従い点滅タイミング制御信号を生

成する。

CPU基板のU7 (PALCE22V10H-15PC/4) のプログラムの処理概要は、U1 (PIC18F452) からの入力信号であるアドレスデータ (AD[0]~[5]) に従い、出力信号のXCS[0]~[9] を選択的に制御する。これにより、FPGA基板へのデータ送信を実現している。各ソフトウェアの詳細リストを「付録」に記載した。

2. 2. 4. 光合成速度の測定方法

OXY-1とLED光源ユニットを組合せ測定システムを構築した (Fig. 2-14 a, b)。外光による測定誤差防止や暗呼吸を促すために、図の構成に加えてチャンバーを囲む遮光カバーも設置した。PCから連続点灯、又は点滅点灯などの点灯条件を設定し、LEDユニットから光を照射しながら、測定チャンバー内の液相酸素濃度データを 1 秒毎に記録し、そのデータを時間で微分することで酸素発生速度 ($\text{nmolO}_2 \cdot \text{ml}^{-1} \cdot \text{min}^{-1}$) を算出した。ここでの酸素発生は、光合成のヒル反応 (日本光合成研究会, 2003d) に由来すると考えられるため、これを光合成速度の指標とした。

但し、「酸素発生」は、呼吸や光呼吸による酸素吸収の影響を受けるため、厳密には、光合成活性と相違する (日本光合成研究会, 2003b)。しかし、既報 (森ら, 2002a) にて「酸素発生」を「光合成活性の指標」として用いており、本章でも便宜的に「酸素発生」を「光合成活性の指標」として用いた。さらに、 C_3 植物では、酸素濃度上昇によるワールブルク効果 (日本光合成研究会, 2003c) のため光合成が阻害されるが、この現象は藻類では認められていない (日本光合成研究会, 2003c)。ユーグレナは藻類であるため、この効果の影響は少ないと思われる。

また、測定精度確保のため光を照射しデータを記録した後は、毎回暗呼吸を促しチャンバー内の溶液中酸素濃度が初期値 ($100 \text{ nmolO}_2 \cdot \text{ml}^{-1}$) に戻るまで待ち、つぎの測定を実施した。この暗呼吸により測定時の光合成産物によるチャンバー内のバイオマス増加を相殺し、結果的にチャンバー内の細胞数をおおよそ一定に保持する様に配慮した。また、測定チャンバーの環境条件は気温 $25 \text{ }^\circ\text{C}$ 、溶液温度 $25 \sim 26 \text{ }^\circ\text{C}$ とした。

PPFD値の測定は、測定チャンバー部分を仮に取り外し、チャンバー部分の定位置に光量子センサを置き測定した。測定器には、光量子センサ (Quantum Sensor LI-190, LI-COR Ltd) 及び、データロガー (LI-1400, LI-COR Ltd) を使用した。

2. 2. 5. 光合成の光利用効率を評価する指標

ユーグレナの効率的な培養を行うためには、ユーグレナにおける光合成の光利用効率を評価する指標が必要である。具体的には、照射した光量 (PPFD) や酸素発生速度から光合成の光利用効率を算出する指標である。そこで、照射した光量 (PPFD) 当たりの酸素発生速度 (光合成速度) を便宜的に「光合成PPFD効率」と呼び、以下の式で定義する。

$$\text{光合成PPFD効率} = \text{酸素発生速度 (nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1}) / \text{PPFD値 (}\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}\text{)} \quad (\text{式2-2})$$

補足だが、類似した用語に「量子収率」という用語がある。「量子収率」は「(光化学反応産物の数) / (吸収された全光量子数)」(Lincoln・Eduardo, 2004) と定義されているが、光合成PPFD効率は「照射した光量子」を用いて定義した。従って、「照射」した光量子の全てが「吸収」されるとは限らないため、両者を区別した。

2. 2. 6. 実験方法

2. 2. 6. 1. 光-光合成曲線の測定方法

初めに「光-光合成曲線」の測定を実施した。コントロールとして連続光を用いPPFD値を広範囲に変化させた場合の光合成速度を測定した。目的は、

- ① PPFD値の変化による光合成速度の光飽和特性などの全体的な光合成速度の傾向を把握することと、
- ②単位PPFD 当りの光合成速度 (光合成PPFD効率) が最大になる条件の傾向を求めるこ

とである。

次に、点滅光を用いPPFD値を変化させた場合の光合成速度を測定した。目的は連続光と比較して点滅光の場合に光合成速度の違いがあるかを調べるためである。測定に用いた点滅光のタイミングをFig. 2-2 に示す。点滅周期 (t_2) を 1.0×10^{-3} s としデューティ比 (t_1 / t_2) を 50 % とした。具体的には点滅周期開始時 ($t = 0$) に連続光時の 2 倍 (デューティ比の逆数倍) のPPFD値で点灯を開始し、周期の 50 % になった時点で消灯するという動作を連続的に繰り返し照射した。こうすることでPPFD値の時間平均値を連続光と同等にした。この条件にてPPFD値 (時間平均値) を連続光と同様に变化させ測定した。以降、このPPFD値の時間平均値を明確に区別する必要がある場合には「PPFD設定値」と記載する。

2. 2. 6. 2. 光-光合成曲線の急峻部分の測定方法

「光合成PPFD効率」が最大になる条件を求めるため、光合成速度が急峻に増加しているPPFD設定値を中心に、測定点数を増やして詳細に測定した。

2. 2. 6. 3. 光合成速度に対する点滅周期の影響の測定方法

「光合成速度に対する点滅周期の影響」を求めるため、点滅光の点滅周期を変化させた場合の光合成速度を測定した。目的は点滅周期の変化が光合成速度へ及ぼす影響を把握するためである。PPFD設定値 (時間平均値) は「2.3.2」に基づき「光合成PPFD効率」が最も高くなるPPFD設定値 ($35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$) とし、デューティ比を 50 % とした。点滅周期 (t_2) を $1.0 \times 10^{-7} \sim 18$ s まで変化させ測定した。

2. 3. 結果

2. 3. 1. 光-光合成曲線

コントロール (連続光) の測定結果は、PPFD設定値が 0 ~ 約 $200 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ までの区間で光合成速度が急峻に増加し約 $6.0 \text{ nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1}$ まで増加した (Fig.

2-15)。さらにPPFD設定値が約 $200 \sim 400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の区間では微増から飽和傾向 (Point-A) に変化し、 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 以上では光合成速度がおおよそ一定となり飽和した。また、点滅光での測定結果と連続光での結果を比較すると、この測定条件では両者の光合成速度は、おおよそ同等であった (Fig. 2-15)。

2. 3. 2. 「光合成PPFD効率」の最大値

「光合成PPFD効率」が最大になる条件を求めるため、図中の原点を通り光合成曲線と接する補助線 (Line-A) を引いた (Fig. 2-16)。ここで、式2-2 の両辺に「PPFD値」乗じて式を変形すると、以下の式2-3 になる。

$$\text{酸素発生速度} = \text{光合成PPFD効率} (\eta) \times \text{PPFD値} \quad (\text{式2-3})$$

式2-3 は、図上で原点を通り傾きが光合成PPFD効率 (η とする) の直線群となる。この直線の中で光合成曲線と交点又は接点を持つ直線を考えた場合、最も傾き (η) が大きくなる直線はC点を通る直線である。従って、補助線と曲線との接点 (Point-C) は光合成PPFD効率 (η) が最大になる点であり、接点のPPFD設定値は約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ だった (Fig. 2-16)。よって、光合成PPFD効率が最大になるPPFD設定値は、約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ という結果であった。

また、点滅光での測定結果と連続光での結果を比較すると、この測定条件では両者の光合成速度は、おおよそ同等であった (Fig. 2-16)。補足だが、光合成曲線と横軸との交点 (Point-B) は光補償点であり、PPFD設定値の実測値は約 $9 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ だった。

2. 3. 3. 光合成速度に対する点滅周期の影響

光合成速度が急に低下する周期を求めるため、図中に補助線を引いた。まず曲線の平坦な部分に重なる補助線を「Line-B」とし、曲線部に重なる補助線を「Line-C」と

した。さらに両補助線の交点を「Point-D」とした。図を観察すると「Point-D」は、光合成速度が急に低下する周期の指標になる。従って、「Point-D」を「光合成速度が急に低下する周期」とした。以降「光合成速度が急に低下する周期」を「変化点」と呼ぶこととする。

図からこの曲線の「変化点」は、点滅周期が約 5.0×10^{-2} s の点であり、この付近から光合成速度が徐々に低下した (Fig. 2-17)。従って、PPFD設定値一定の条件下でも点滅周期を約 5.0×10^{-2} s 以上にすると、それ以下の場合と比較して光合成速度が低下するという結果になった。

2.4. 考察

本章の目的である「点滅光での光-光合成曲線」や「光合成速度に対する点滅周期の影響」などのデータを得ることが出来た。得られたデータの特長をまとめると、①連続光条件でPPFD設定値を 0 から増加させた場合、約 $200 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ までは光合成速度が急増するが、これ以降は微増傾向になり約 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ で飽和した (Fig. 2-15)。より正確に光飽和点を算出する方法として「光合成曲線の近似式」から算出する方法があるが、植物の種類により適合する近似式は異なる。そこで、「ユーグレナに適合した光合成曲線の近似式」を文献などにて探したが見つからなかった。従って、ここでは連続光での実測値 (Fig. 2-15) から光飽和点を約 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ と読み取った。

②連続光条件でPPFD設定値当たりの光合成速度(光合成PPFD効率)が最大になるPPFD設定値は、約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ だった (Fig. 2-16)。

③点滅光条件で点滅の周期を 1.0×10^{-3} s としデューティ比を 50% 一定としてPPFD設定値を変化させた場合の光合成速度は、連続光とおおよそ同じだった (Fig. 2-15, 2-16)。

④点滅周期を変化させた場合、点滅周期を約 5.0×10^{-2} s よりも長時間にするにつれ、短時間の場合と比較して光合成速度が低下した (Fig. 2-17)。

点滅周期が長くなると光合成速度が低下するという現象は、既報(森ら, 2002a)の OXY-1 を使用した測定結果でも見られ、この低下現象が発生する点滅周期は 10 ms とのことである。一方、本章の実験では 5.0×10^{-2} s (50 ms) と約 5 倍も長い点滅周期で現象が発生したため、大きな違いがあった (Fig. 2-17)。

また、既報(森ら, 2002a)では点滅周期を 400 μ s とした場合、連続光と比較して光合成速度が約 23% 増大したとのことだが、本章の実験では既報の様に光合成速度が増大する現象は現れなかった (Fig. 2-17)。これらの違いの要因は、恐らく、両者で用いた試料の光合成機構の違いが関係するのではないかとと思われる。既報(森ら, 2002a)では試料にサラダナを用いており、光合成反応過程は「チラコイド反応(光化学系 I, II 等)」と「カルビン・ベンソン回路」と思われる。既報(森ら, 2002a)では点滅周期 400 μ s で光合成速度が増大した要因として「光化学系 II の反応中心クロロフィル (P680) の還元時間が 200 μ s で律速条件となり、点滅周期 400 μ s とした場合のパルス幅 (200 μ s) と一致すること」と述べている。

一方、ユーグレナにも光化学系 I と II は存在するが、高等植物と比較すると以下の違いが知られている。

- ①光合成単位(光合成を行う最小単位)が高等植物では、約 2400 分子だがユーグレナは約 1600 分子と少ない(横田, 1989)。
- ②葉緑体中のクロロフィル a/b 比が高等植物では約 2 ~ 3 だがユーグレナは約 5 ~ 7 と高い(横田, 1989)。
- ③チラコイド内でのクロロフィル濃度が高等植物では約 30 mM だがユーグレナは約 300 mM であり約 10 倍も高い(横田, 1989)。

ここで、ユーグレナの光合成単位が高等植物よりも少ない理由として、光合成単位のクロロフィル b 含量が少なく、クロロフィル a/b 比が高等植物より高いことが関係するかもしれない(横田, 1989)。また、一般的にクロロフィル a/b 比は光環境により変化し、例えば高照度で生育したドナリエラ (*Dunaliella*) はクロロフィル a/b 比が増加し 10 を超える場合がある(日本光合成研究会, 2003a)。

これらから類推すると恐らく、ユーグレナのクロロフィルa/b比が高いことでドナリエラの例と同様に高照度（高PPFD値）条件に適合した光合成機構を有するものと思われる。さらに、チラコイド内のクロロフィル濃度が高等植物より10倍も高いため、結果的にサラダナとユーグレナでは光合成反応の「律速段階」が異なったため、本章の結果（Fig. 2-17）が既報（森ら，2002a）と異なる結果になったのではないかと推測される。補足だが、サラダナよりも生物学的な分類上で極力ユーグレナに近い植物種での測定データと比較したいところだが、LED点滅光照射下での植物栽培に関する報告は、サラダナ以外にはなかった。

本章の実験では点滅光のデューティ比を50%に固定した実験だったが、既報（森ら，2002a）では点滅光のデューティ比を変化させると成長率も変化したとのことなので、ユーグレナに関しても点滅光のデューティ比を変化させれば、さらなる光合成効率向上の糸口が得られるかもしれない。

2. 5. まとめ

この結果を産業上の利用面から考えると、例えば赤色LEDを用いてユーグレナ用の効率的な光源装置を設計する場合、実験結果から、PPFD設定値を「約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 」とすれば、光合成PPFD効率が高くなることが分かった。さらに、赤色LED点滅光を照射する場合には、点滅周期を「約 $5.0 \times 10^{-2}\text{s}$ よりも短時間」とすれば、光合成PPFD効率が低下しないことも分かった。これらの結果は、今後ユーグレナ用の高効率な照明システムを開発する上で非常に有益な情報と思われる。

補足だが、ここでは、照明用電力エネルギー削減に着眼し「光合成の光利用効率向上」を目指した。しかし、極端な例として、宇宙船内などの様に利用可能な体積が小さく限定されている状況下で、より多くの収量（食糧）を得たい場合も考えられる。この場合には、光量を光飽和点付近まで増加させて培養することで「培養槽の体積当たりの収量」を増加させるという選択肢も考えられる。従って、実際の培養施設では状況や目的に応じて照射する光量等を選択する場合もあると推測する。

一方、本章ではデューティ比を 50 % に固定した実験のため、デューティ比による影響は分からなかった。以降の実験では、デューティ比も変化させて、より高効率な光合成条件を目指して実験を進めたい。

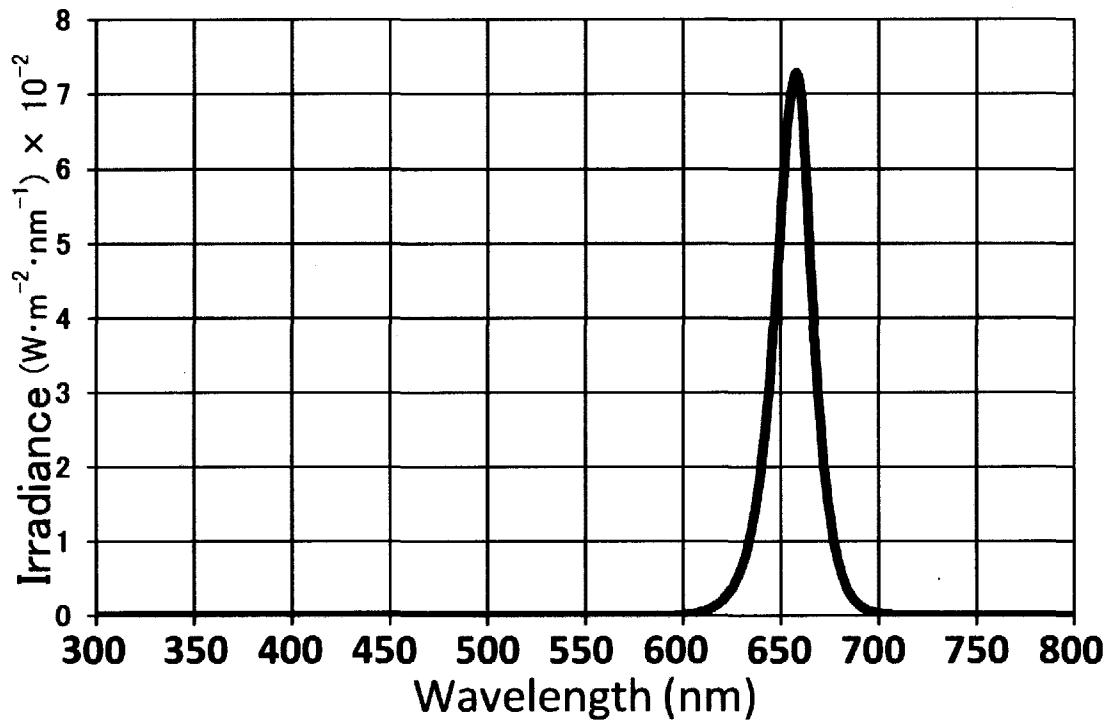


Fig. 2 - 1 Spectral distribution of red LED.

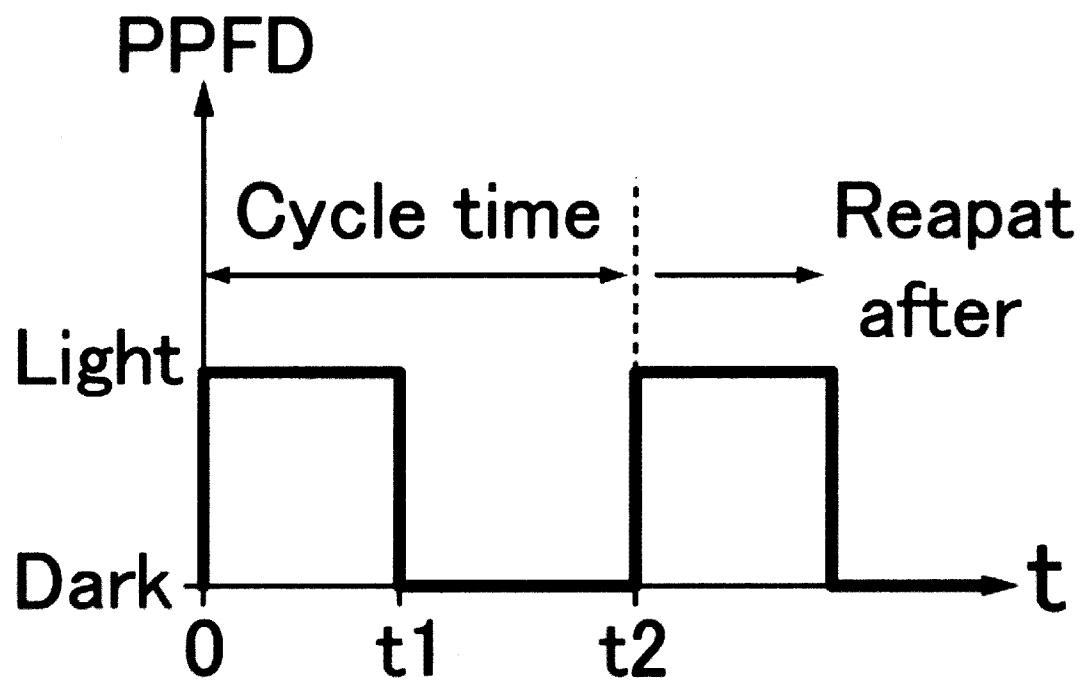


Fig. 2 - 2 LED blink timing.
(Duty ratio = t_1 / t_2)

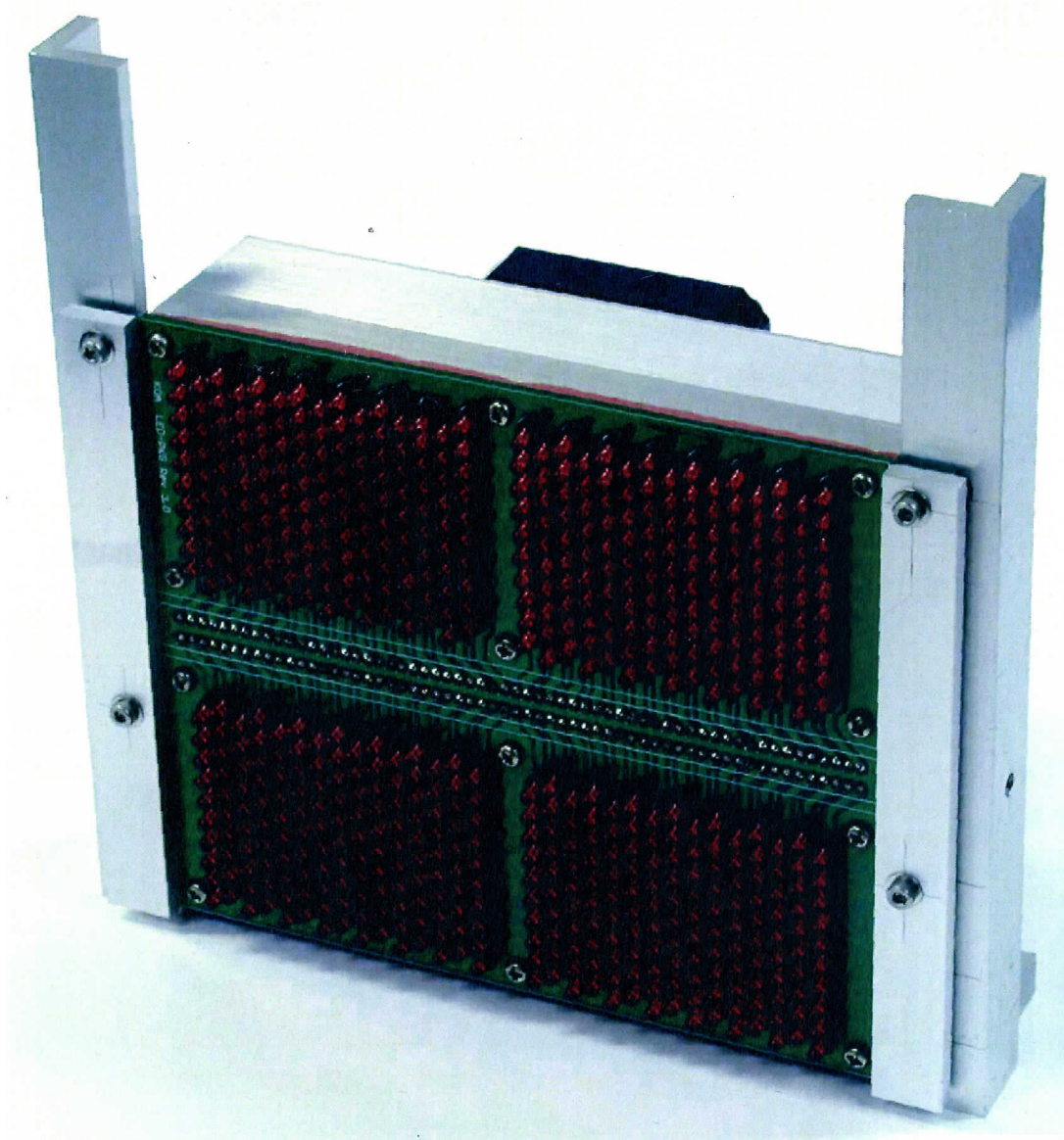


Fig. 2 - 3 The LED panel.
(a) Front side.

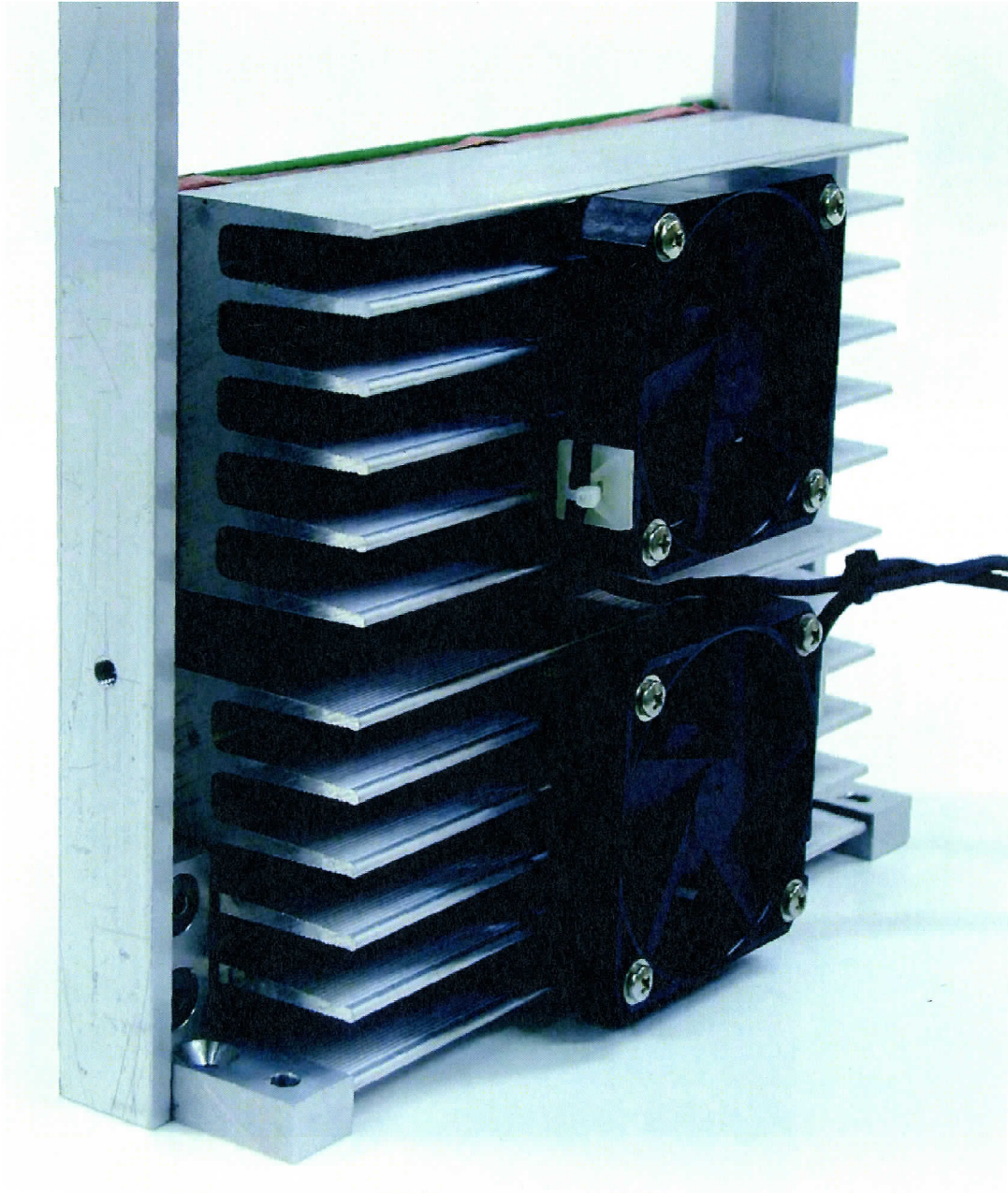


Fig. 2 - 3 The LED panel.
(b) Back side.

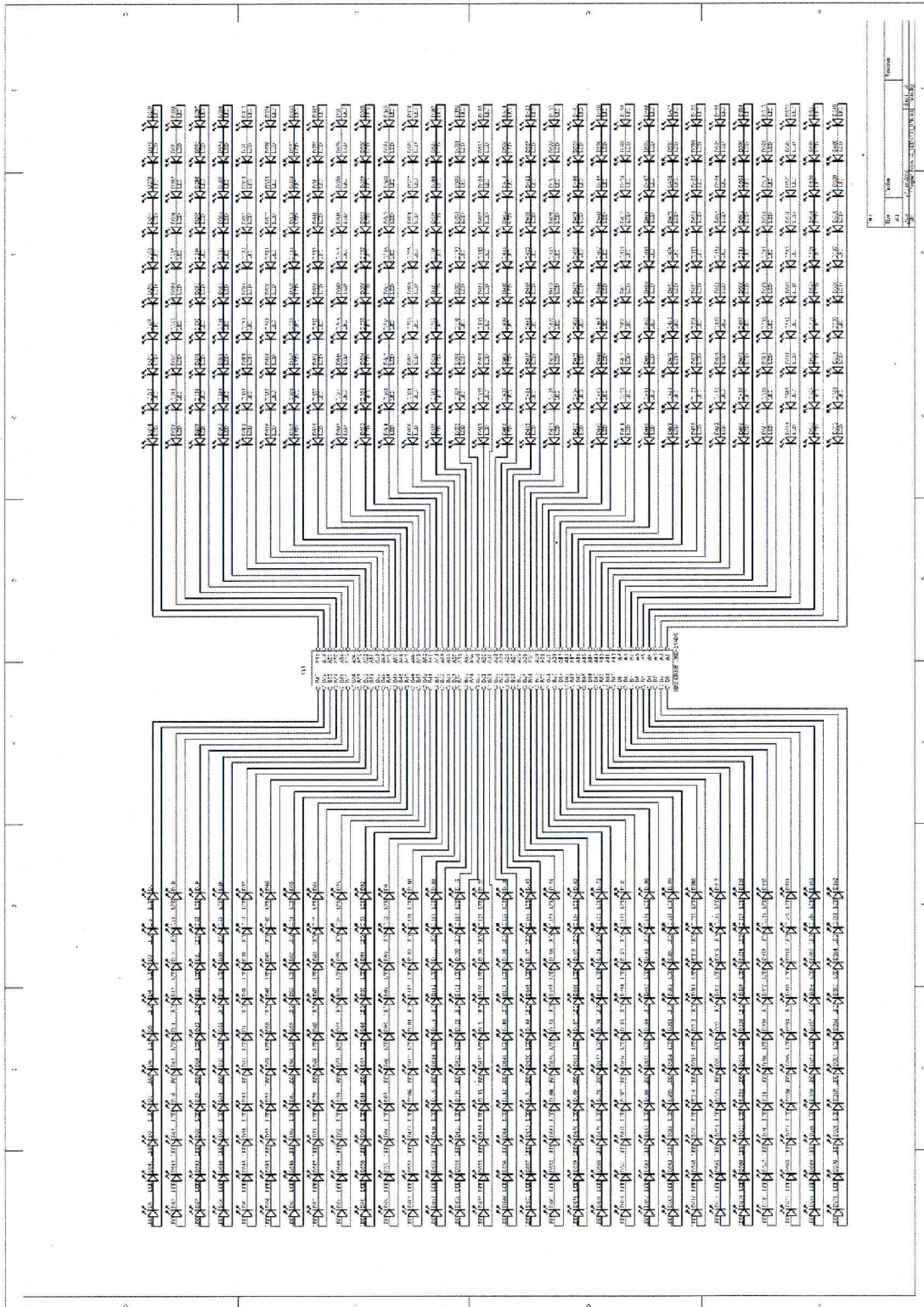


Fig. 2 - 4 The circuit diagram of LED printed-circuit board.

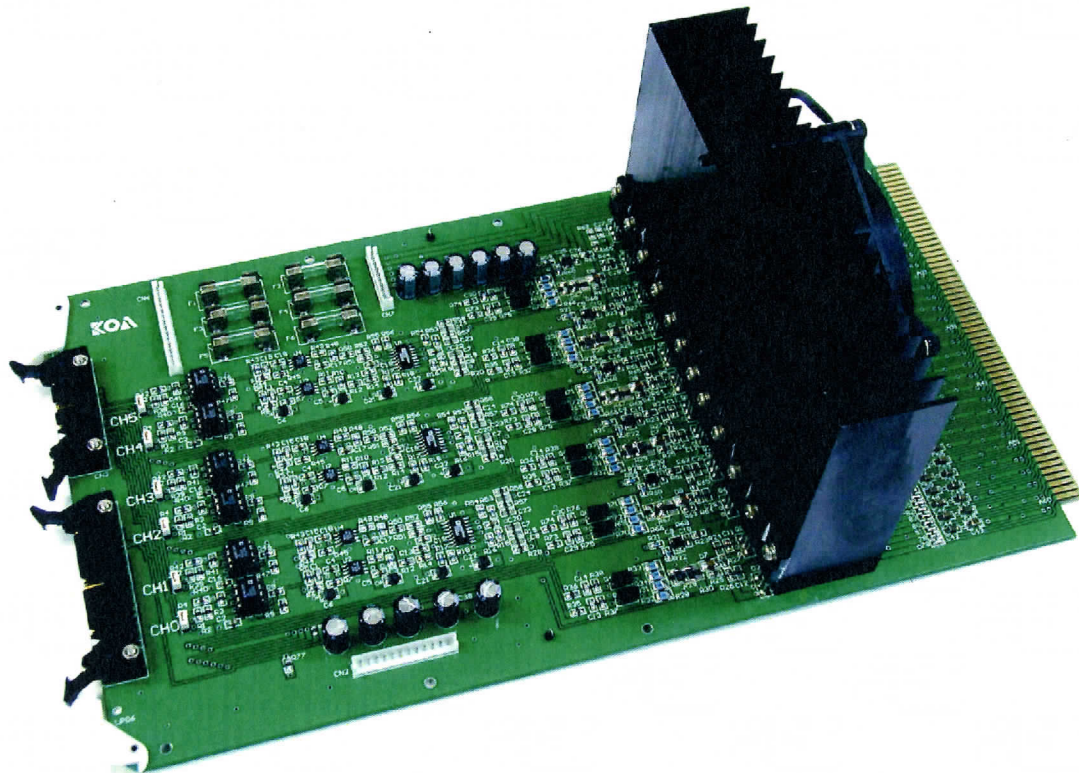


Fig. 2 - 5 The LED drive printed-circuit board.
(320×180 mm)

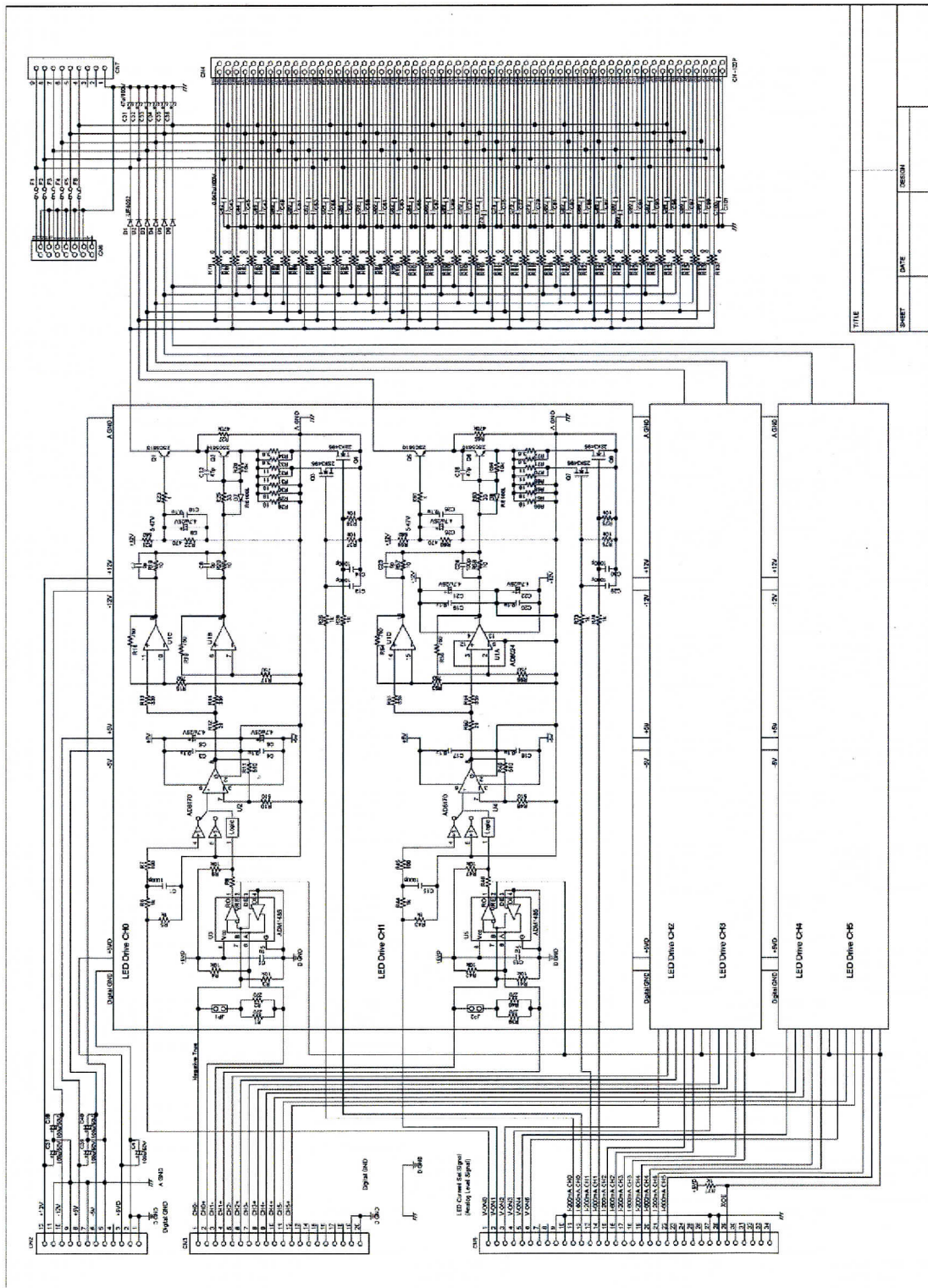


Fig. 2 - 6 The circuit diagram of LED drive printed-circuit board.

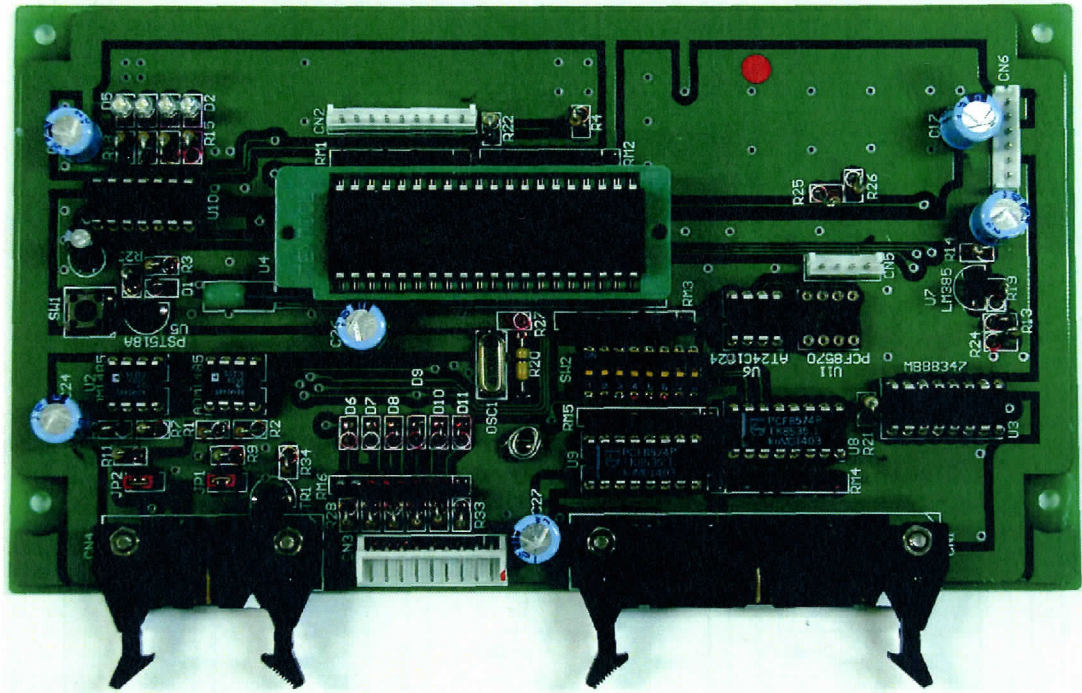


Fig. 2 - 7 The LED current control printed-circuit board.
(180×100 mm)

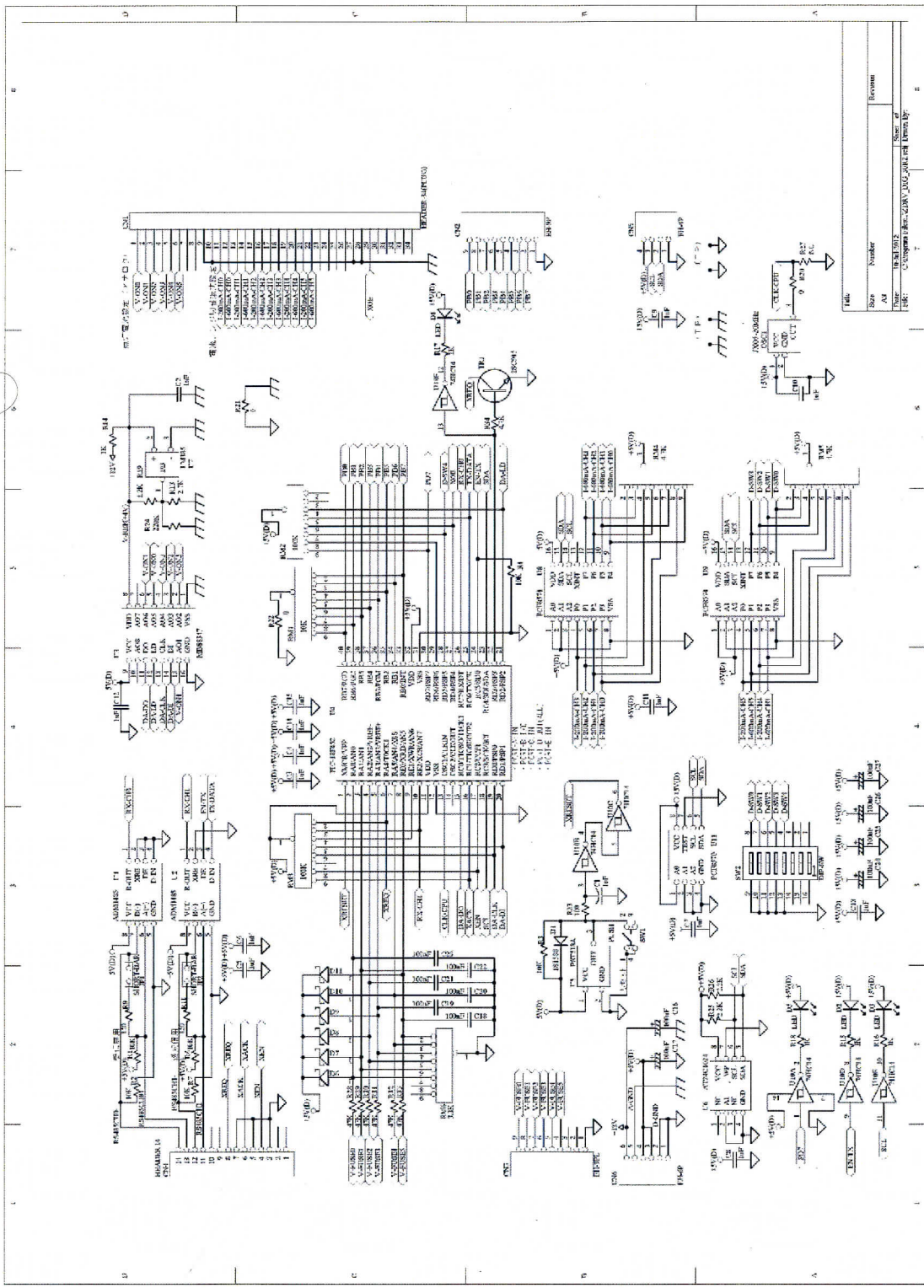


Fig. 2 - 8 The circuit diagram of LED current control printed-circuit board.

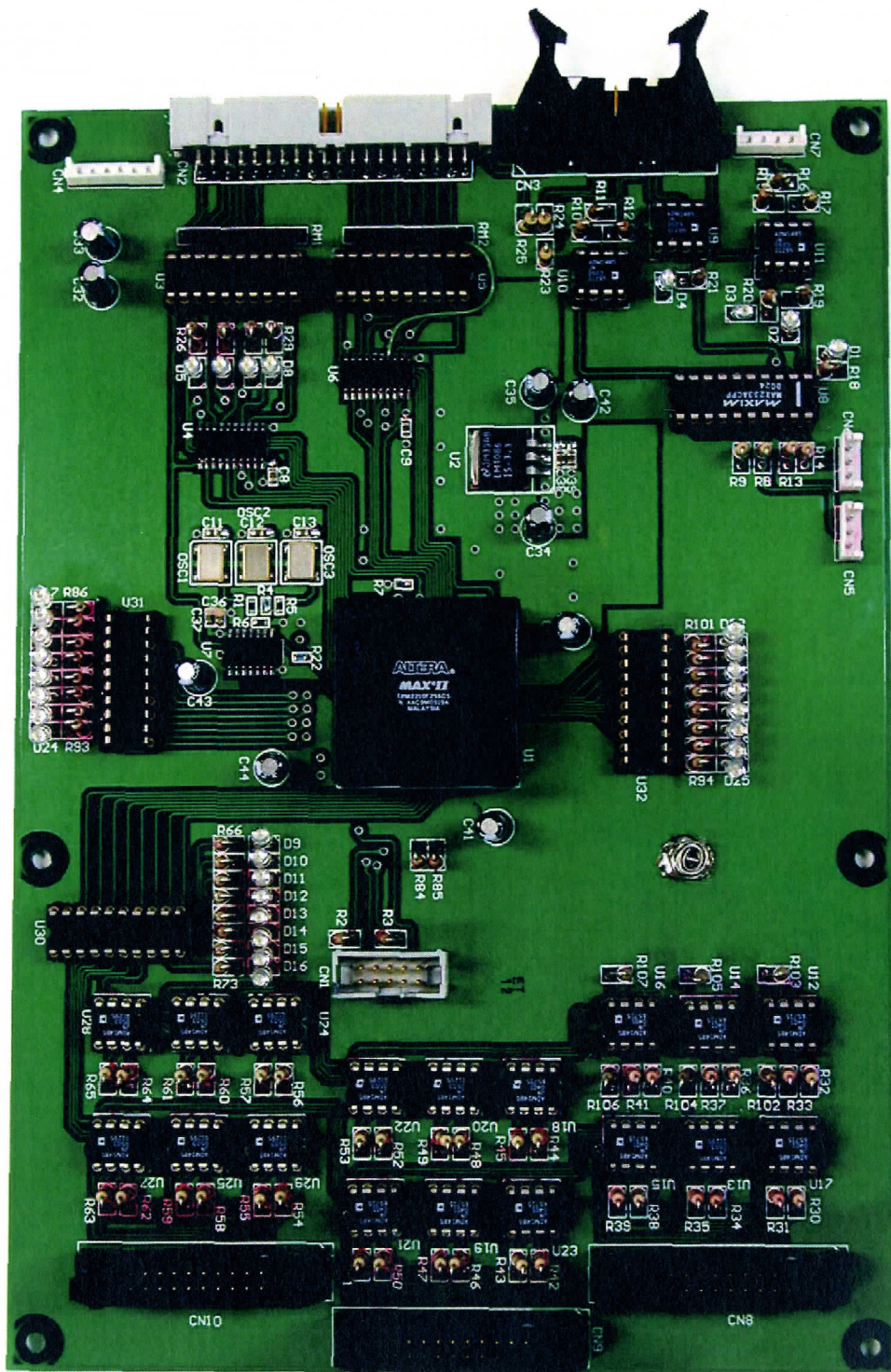


Fig. 2 - 9 The FPGA printed-circuit board.
 (230×160 mm)

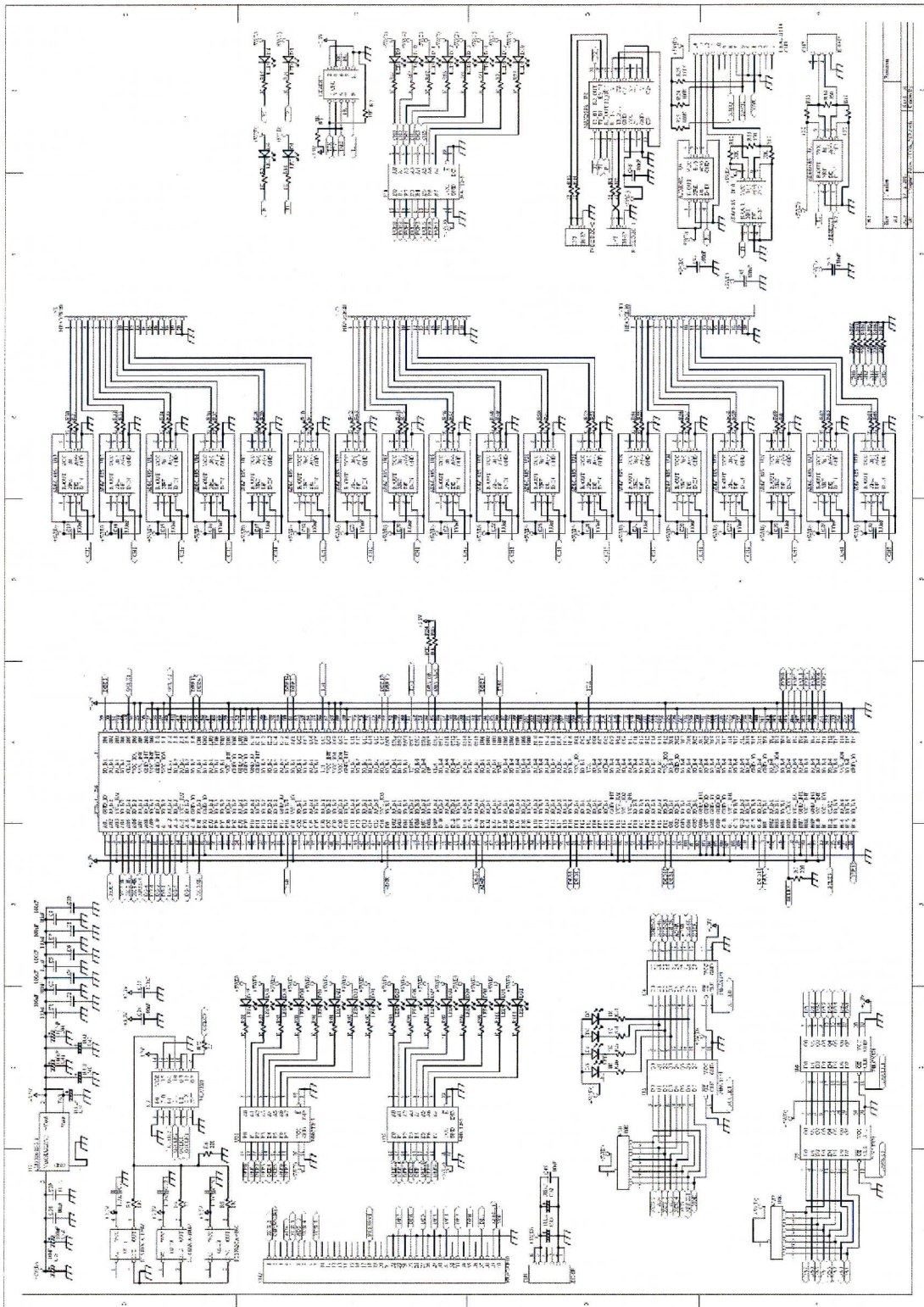


Fig. 2 - 10 The circuit diagram of FPGA printed-circuit board.

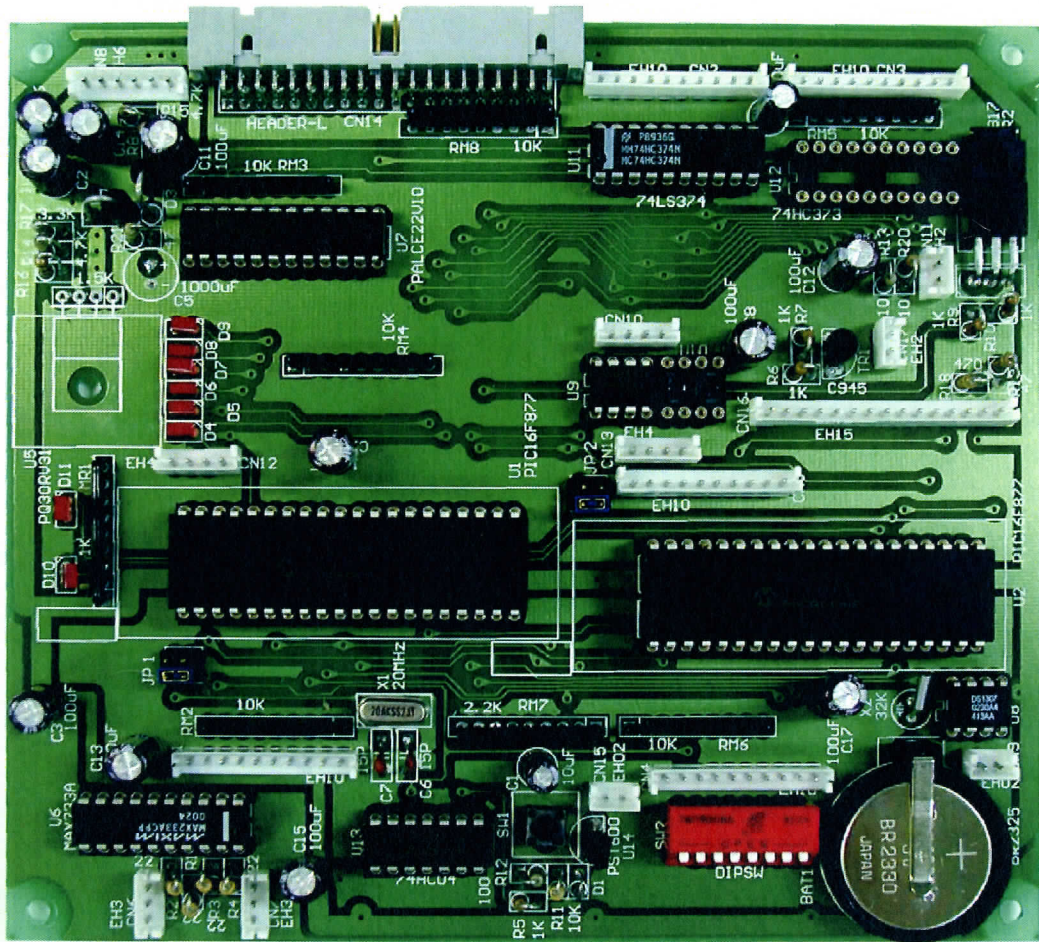


Fig. 2 - 11 The CPU printed-circuit board.
(160×145 mm)

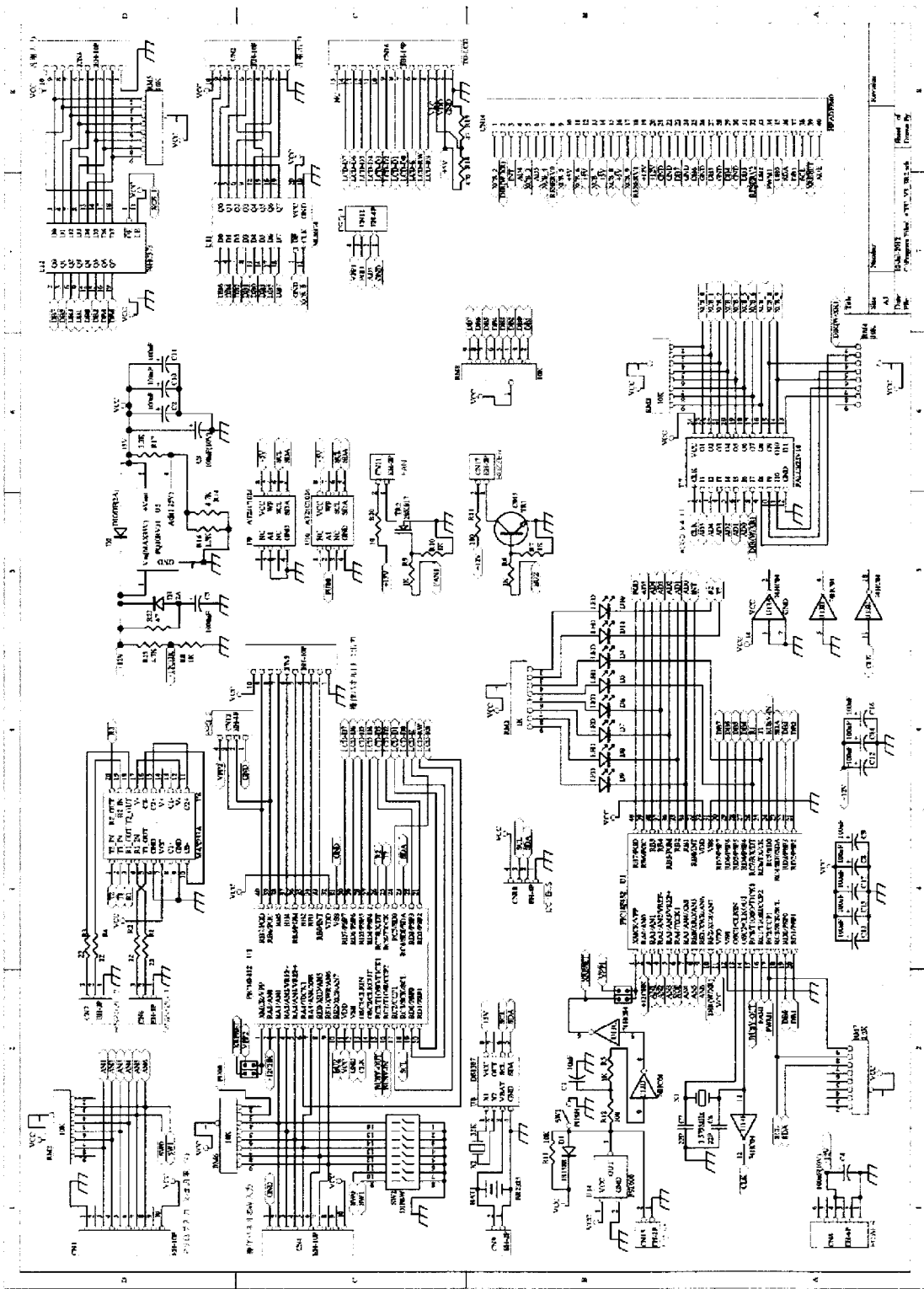


Fig. 2 - 12 The circuit diagram of CPU printed-circuit board.

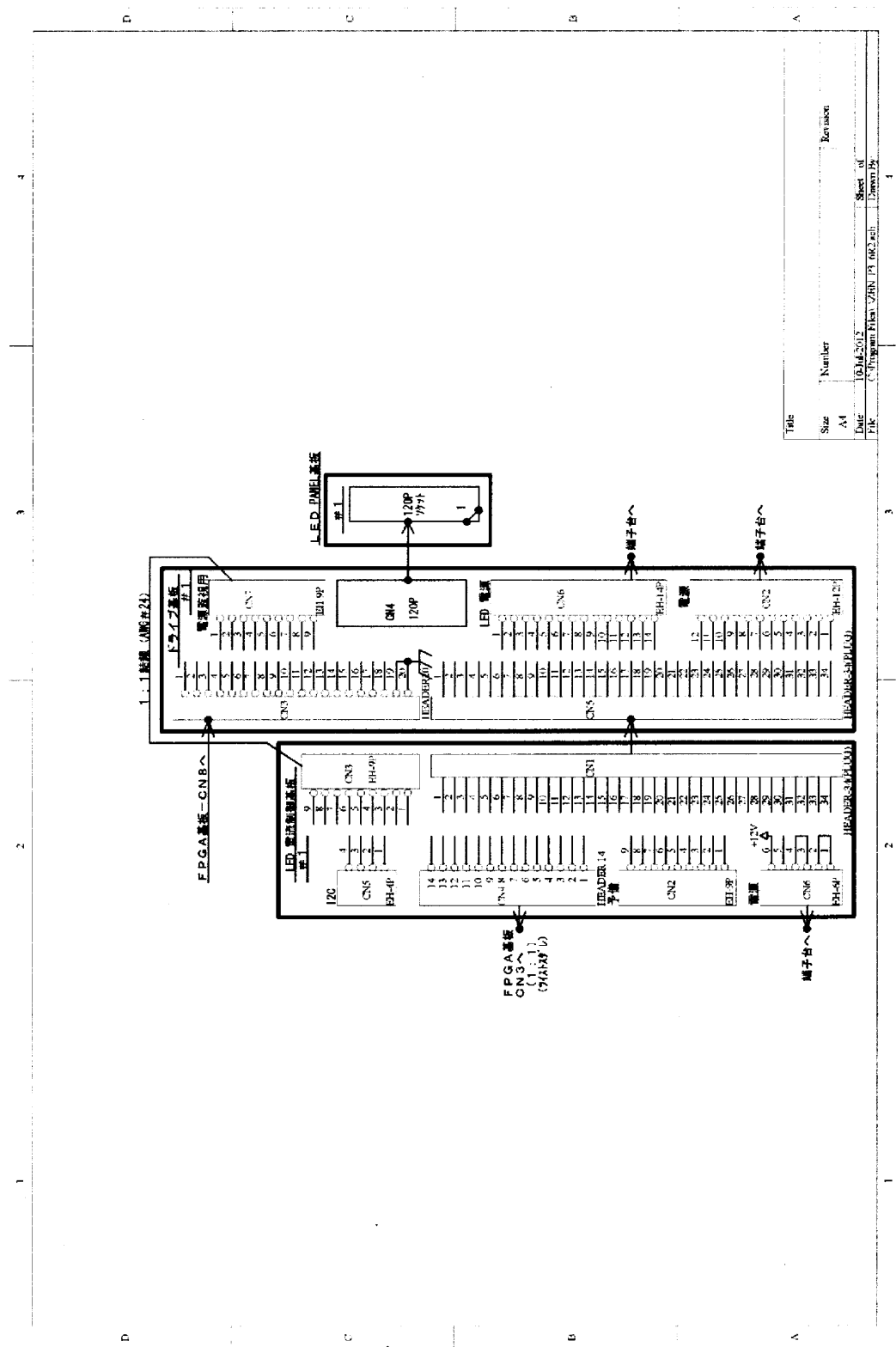


Fig. 2 - 13 The overall wire connection. (a)1/3

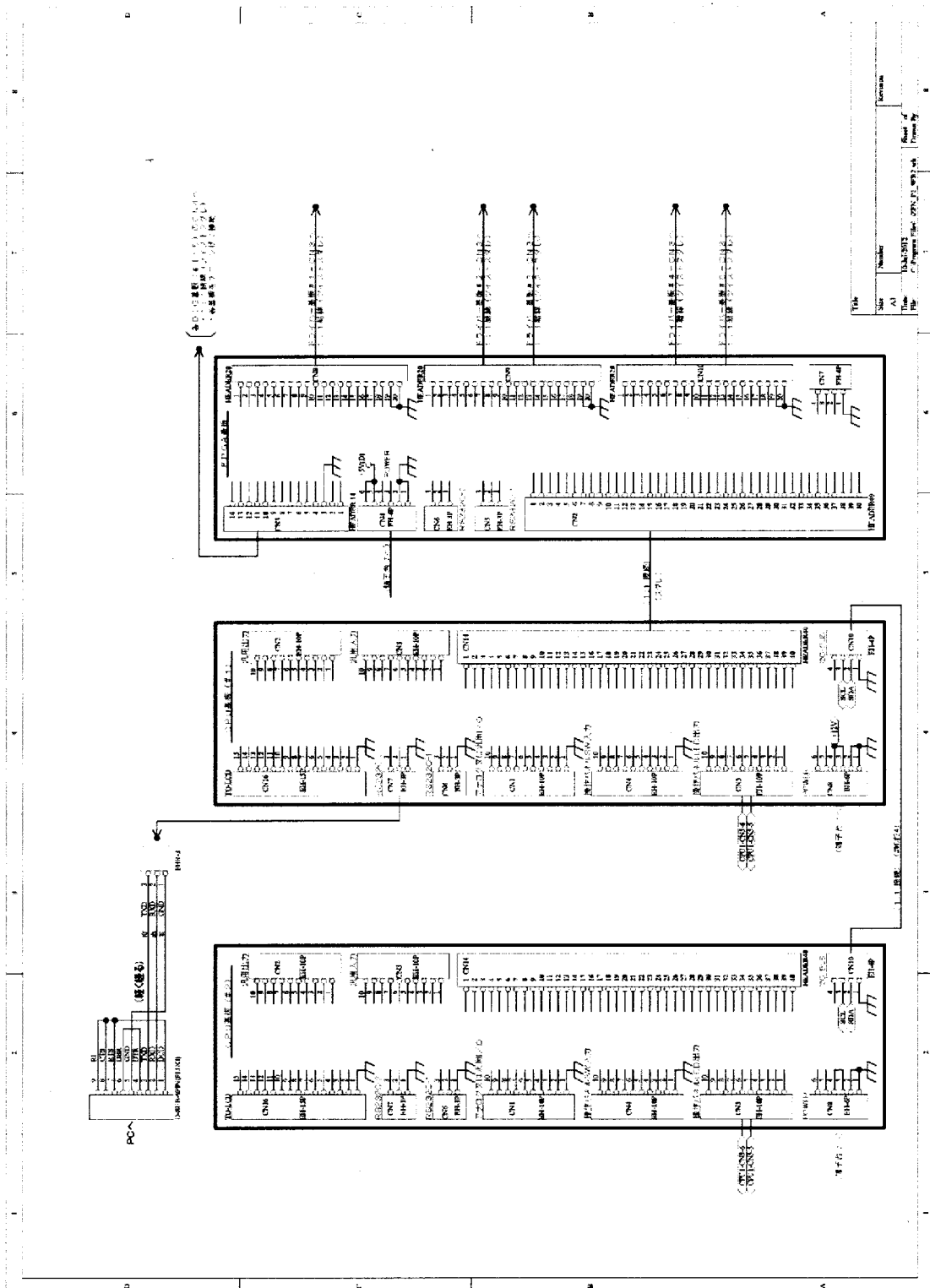


Fig. 2 - 13 The overall wire connection. (b)2/3

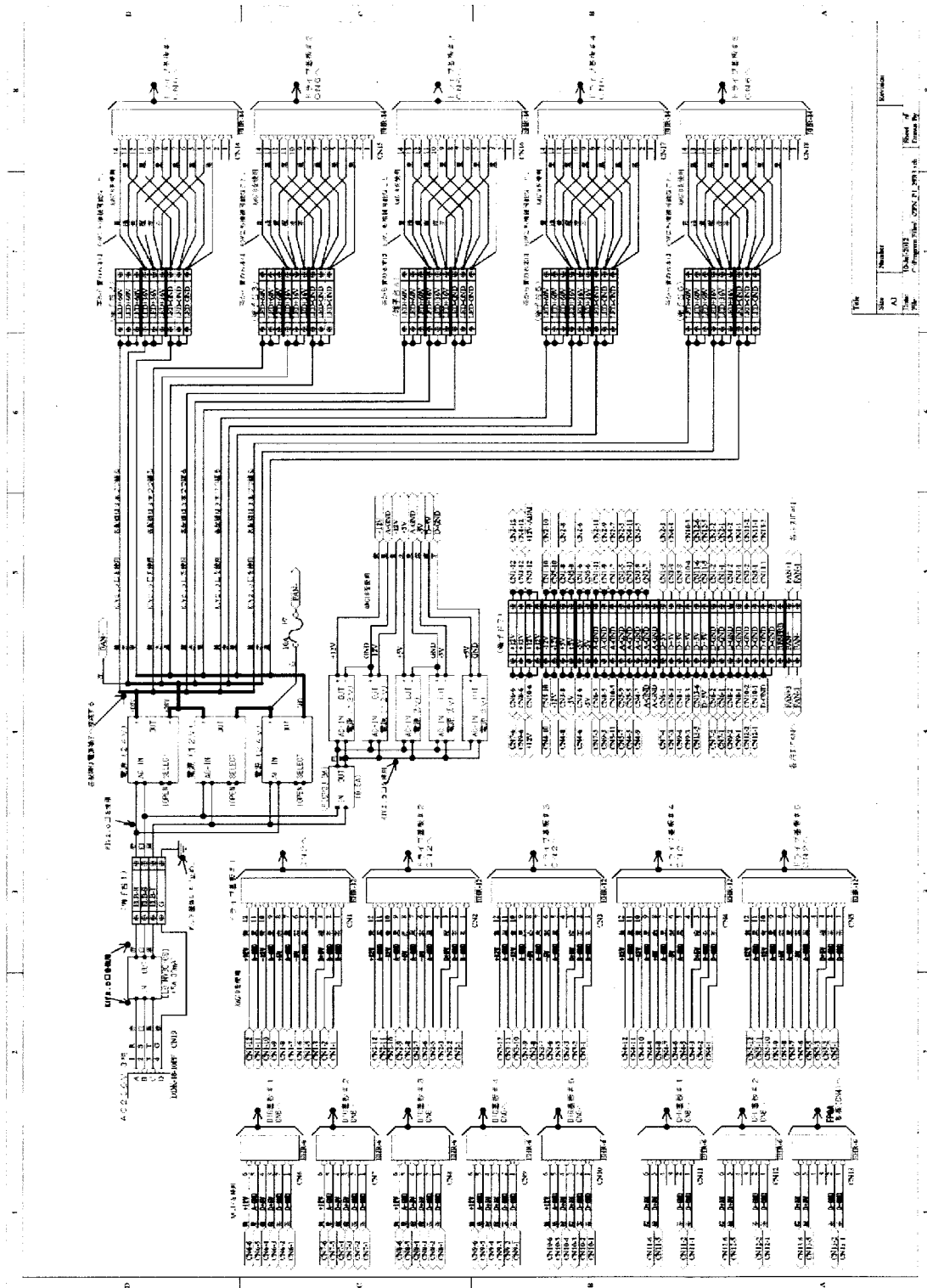


Fig. 2 - 13 The overall wire connection. (c)3/3

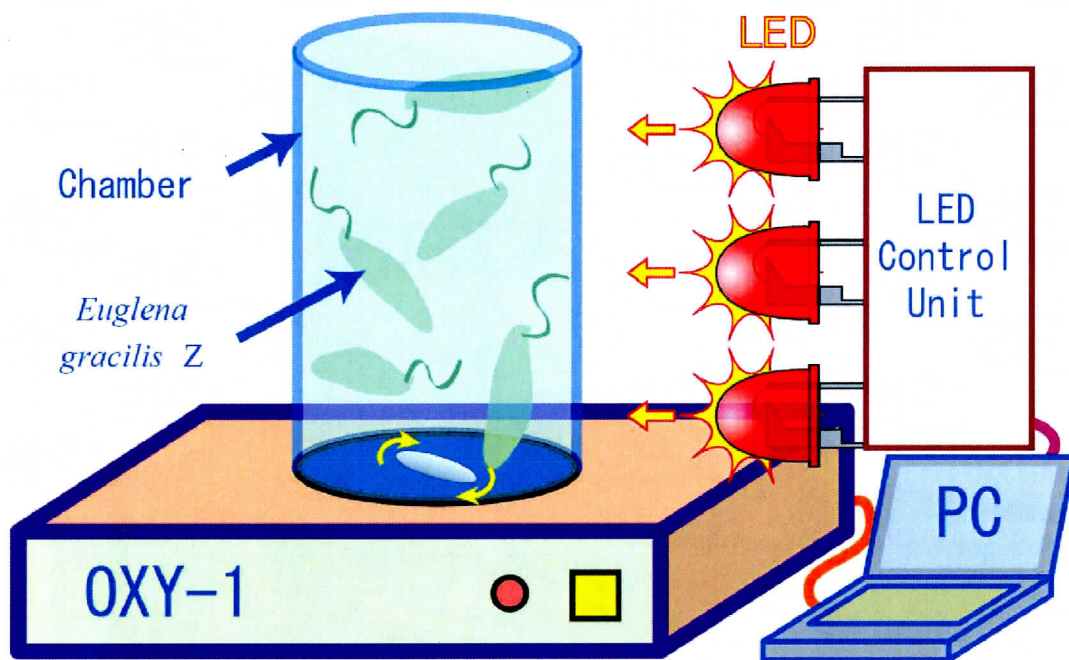


Fig. 2 - 14 The equipment used to measure the photosynthesis rate.
(a) The overall view.



Fig. 2 - 14 The equipment used to measure the photosynthesis rate.
(b) OXY-1 and LED panel.

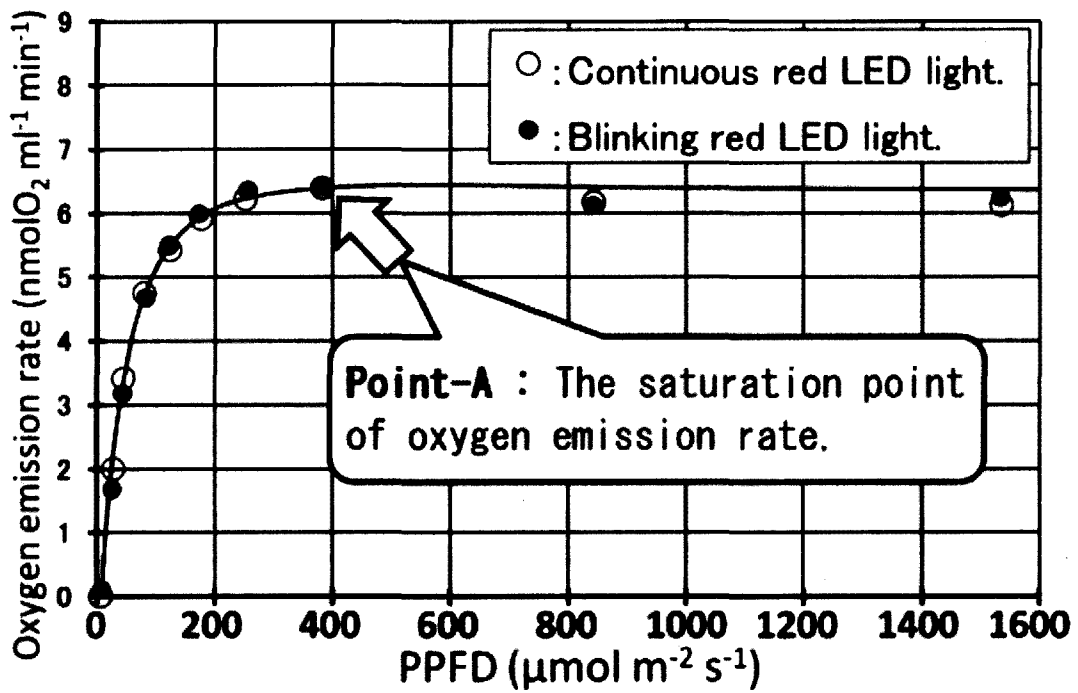


Fig. 2 - 15 Effects of PPFD on the oxygen emission rate under continuous and blinking red LED light.

○: Continuous red LED light
 ●: Blinking red LED light

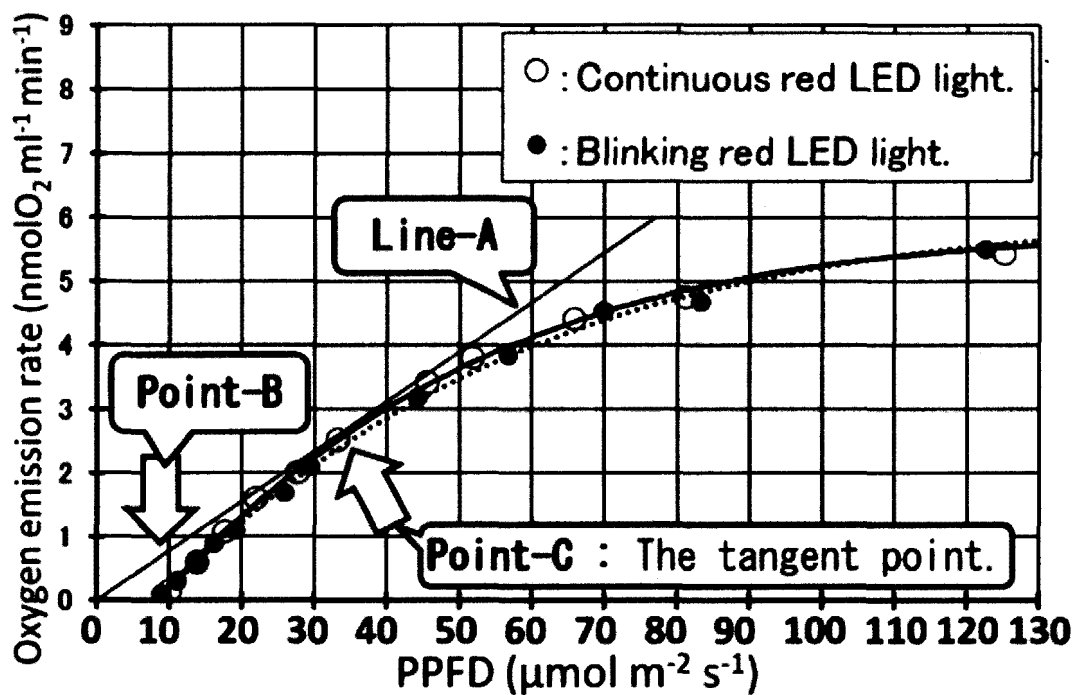


Fig. 2 - 16 Effects of PPFD on the oxygen emission rate under continuous and blinking red LED light.

○: Continuous red LED light

●: Blinking red LED light

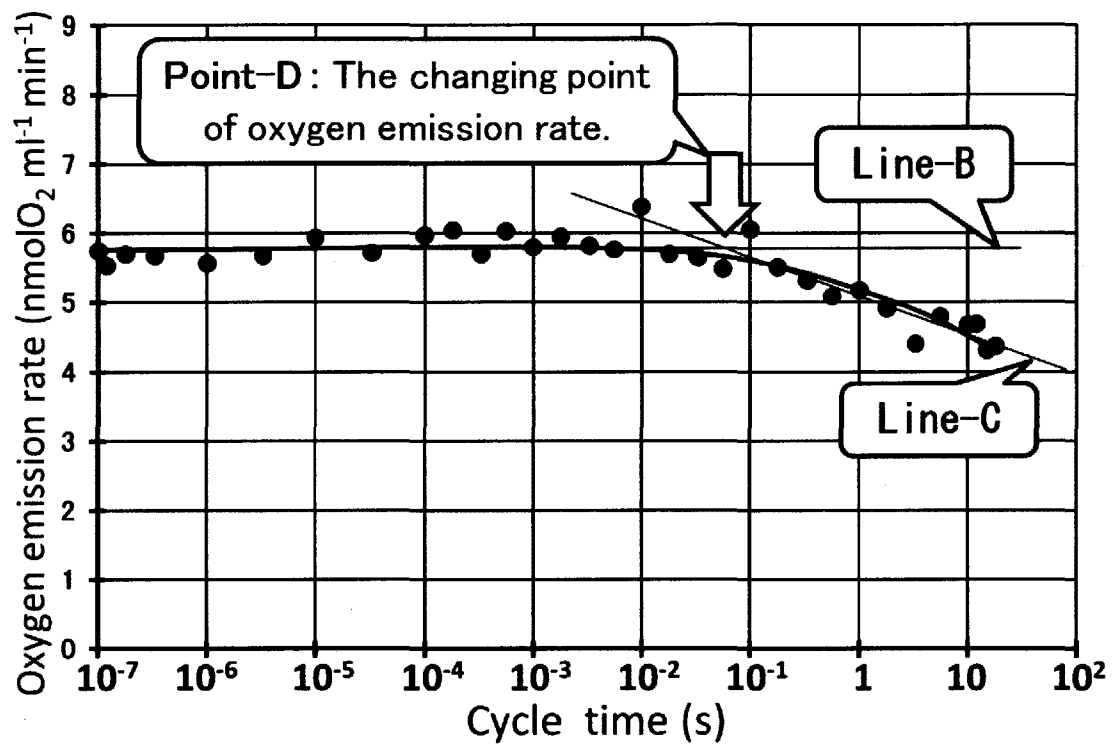


Fig. 2 - 17 Effects of the cycle time of blinking red LED light on oxygen emission rate for $\text{PPFD} = 35 \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$.

Table. 2 - 1 Culture medium for *euglena*.

	Chemical compound	concentration	unit
1	$(\text{NH}_4)_2\text{HPO}_4$	1.0	gL^{-1}
2	KH_2PO_4	1.0	gL^{-1}
3	MgSO_4	0.2	gL^{-1}
4	$\text{Na}_3(\text{C}_3\text{H}_5\text{O}(\text{COO})_3) \cdot 11\text{H}_2\text{O}$	0.8	gL^{-1}
5	CaCl_2	20	mgL^{-1}
6	$\text{Fe}_2(\text{SO}_4)_3 \cdot 6\text{H}_2\text{O}$	3.0	mgL^{-1}
7	$\text{MnCl}_2 \cdot 4\text{H}_2\text{O}$	1.8	mgL^{-1}
8	$\text{Co}(\text{NO}_3)_2 \cdot 6\text{H}_2\text{O}$	1.3	mgL^{-1}
9	$\text{ZnSO}_4 \cdot 7\text{H}_2\text{O}$	0.4	mgL^{-1}
10	H_2MoO_4	0.2	mgL^{-1}
11	$\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$	20	μgL^{-1}
12	Thiamine hydrochloride	10	μgL^{-1}
13	Cyanocobalamine	0.5	μgL^{-1}

第3章

赤色LED点滅光の点滅周期とデューティ比
がユーグレナの光合成速度に
及ぼす影響

3. 1. 序論

第 2 章にて、ユーグレナにおける赤色LED点滅光での「光-光合成曲線」や「連続光と点滅光での光合成速度の比較」、「光合成PPFD効率を最大にする光量条件」、「光合成速度に対する点滅周期の影響」を調査した。その結果、ユーグレナにおける点滅光及び連続光での「光-光合成曲線」や光合成PPFD効率を最大にする光量(PPFD値)、最適な点滅周期を明らかにした。しかし、第 2 章では点滅光のデューティ比 = 50 % の条件下のみで光合成速度を測定したため、点滅光のデューティ比による影響は、分からなかった。

植物に照射する光の点滅デューティ比を最適化することで、植物の成長率などを向上させた報告例がある。具体的には、既報(森ら, 2002a)にて点滅光のデューティ比を 33 % にした場合に、連続光と比較してサラダナの成長率が約 25 % 増大したと報告された。従って、筆者らは、ユーグレナに照射する赤色LED点滅光のデューティ比を様々な値に変更しながら光合成速度を測定した。

さらに第 2 章では、「光合成速度に対する点滅周期の影響」を調べる際に、光量(PPFD値)を $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の条件下のみで測定したため、光量の増減による影響は分からなかった。光量と点滅周期の光合成速度に対する相互依存性を確認することは、今後高効率なユーグレナ用照明システムを開発する際に、照明システム設計の根幹に関わる重要な事項である。従って、筆者らは、光量を段階的に変更しながら、ユーグレナにおける「光合成速度に対する点滅周期の影響」を調べた。上述の調査内容の詳細を以下に報告する。

3. 2. 材料及び実験方法

3. 2. 1. 試料と培養方法

測定用試料には、ユーグレナ(*Euglena gracilis* Z 株)を培養して用いた。培養方法は、「2.2.1.」に従った。

3. 2. 2. 試料の調整方法

試料の調整方法や、測定チャンバーへ試料を注入する方法などの手順は、「2.2.2.」に従った。

3. 2. 3. LED光源の仕様

実験に用いたLED光源ユニットの仕様や、使用したLED素子や、LED光源の制御方法は、「2.2.3.」に従った。

3. 2. 4. 光合成速度の測定方法

「2.2.4.」に従い、クラーク電極タイプの液相酸素濃度測定器 (OXY-1, Hansatech Instruments Ltd) とLED光源ユニットを組合せて測定システムを構築した。外光による測定誤差防止や試料の暗呼吸を促すために、チャンバーを囲む「遮光カバー」も設置した。LED光源ユニットの点滅点灯の制御方法や液相酸素濃度の測定手順や光合成速度の算出方法やPPFD値の設定方法や周囲環境条件は、「2.2.4.」に従った。

さらに、液中細胞濃度の増減による影響を正規化するため、細胞数当たりの酸素発生量から光合成速度を算出した。また、チャンバー内における光の減衰率を見積もると、液中細胞濃度が約 $1.0 \times 10^{-5} \text{ cell} \cdot \text{ml}^{-1}$ と低密度であり、さらに測定チャンバーの直径が約 $1.0 \times 10^{-2} \text{ m}$ と短いため、チャンバー内における透過光の減衰率を算出(増田ら, 2006) すると、0.76% であり僅かな数値であった。従って、ここではチャンバー内における透過光の減衰率を無視し、チャンバー内の光量は均一であるとして扱った。

3. 2. 5. 実験方法

3. 2. 5. 1. 各PPFD値における、光合成速度に対する点滅周期の影響を測定

測定に用いた「点滅光のタイミング波形」や「PPFD設定値」の調整方法は、「2.2.6.1.」

に従った。試料に照射する点滅光のPPFD設定値を 35、64、118、218、400 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ として各PPFD設定値についての光合成速度を測定した。目的は、PPFD設定値の違いによる光合成曲線の違いを比較するためである。点滅周期は $1.0 \times 10^{-6}\text{s} \sim 10\text{s}$ まで変化させ、デューティ比は 50 % とした。

3. 2. 5. 2. デューティ比の影響を測定

LED点滅光のデューティ比を変化させた場合の光合成速度を測定した。目的は、デューティ比の変化が光合成速度へ及ぼす影響を調査することである。既報(森ら, 2002a)にて、点滅周期を $4.0 \times 10^{-4}\text{s}$ に設定した場合にサラダナの光合成速度が増大したことで、成長率が増加したとの報告がある。従って、ユーグレナでも同様な光合成速度の増加現象が再現するか否かを確認するため、同じ点滅周期条件 ($4.0 \times 10^{-4}\text{s}$) を追加した。

測定に用いた点滅光のタイミングをFig. 2-2 で説明すると、まず点滅周期(t_2)を $1.0 \times 10^{-3}\text{s}$ とし、 t_1 を変化させることでデューティ比を 100 ~ 10 % まで変化させて光合成速度を測定した。具体的には、例としてデューティ比 = 10 % の場合、点滅周期開始時刻($t = 0$)でPPFD設定値(時間平均値)の 10 倍(デューティ比の逆数倍)のPPFD値で点灯を開始し、経過時間が周期の 10 % (デューティ比の設定値)になった時点で消灯するという動作を連続的に繰り返すという方法で点滅光を照射した。この方法で各デューティ比でのPPFD設定値(時間平均値)が常に $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ になる様にし、点滅光を試料に照射した。さらに、点滅周期(t_2)を $1.0 \times 10^{-1}\text{s}$ 及び、 $4.0 \times 10^{-4}\text{s}$ として上述と同様にデューティ比を変化させて光合成速度を測定した。

3. 3. 結果

3. 3. 1. 各PPFD値における、光合成速度に対する点滅周期の影響

PPFD設定値 = 35、64、118、218、400 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ での結果を図にプロットした(Fig. 3-1)。例として、PPFD設定値 = 400 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の曲線を観察すると、点滅周期を

$1.0 \times 10^{-6} \sim$ 約 1.0×10^{-2} s (Point-F)までの範囲にした場合、光合成速度は、おおよそ変化せず一定の値だった。しかし、点滅周期を Point-F よりも長時間にすると光合成速度は、急に低下した。以降、「光合成速度が急に低下する周期」を「変化点」と呼ぶこととする。

従って、PPFD設定値 = $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ での変化点(Point-F)の周期は約 1.0×10^{-2} s である。一方、PPFD設定値 = $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ での変化点(Point-E)の周期は、約 5.0×10^{-2} s であり Point-F と比較すると Point-E の方が長時間であった(Fig. 3-1)。他のPPFD設定値条件での各曲線に関してもおおよそ同様の傾向があった。よって、PPFD設定値を大きくするに従い、各曲線上の変化点(Point-F, G, H, J)の図上での位置は、短時間方向へ変化した(Fig. 3-1 の破線を参照)。

3. 3. 2. デューティ比の影響

図で光合成速度の各線を比較すると、点滅周期を 1.0×10^{-3} s 及び、 4.0×10^{-4} s とした場合には、両者の各線は、おおよそ重なっていた(Fig. 3-2)。さらに、デューティ比を 100 ~ 10 % まで変化させても、両者の光合成速度は、おおよそ変化せず一定の値だった。一方、点滅周期を 1.0×10^{-1} s にした場合には、デューティ比を下げるに従い光合成速度も低下する傾向があった。特に、デューティ比を 10 % (Point-K) にした場合に最も光合成速度が低下した(Fig. 3-2)。補足だが、図の右側に記載がある「デューティ比 = 100 %」の意味は「連続光照射」と同義である。

3. 4. 考察

本章の目的である「点滅周期やデューティ比を変化させた場合の、光合成速度に対する影響のデータ」を得ることが出来た。得られたデータの特長をまとめると、
①PPFD設定値の条件を、35、64、118、218、 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の各条件毎に、光合成速度と点滅周期の関係を比較すると、図の破線で示した様にPPFD設定値を大きくするに従い、各曲線の変化点の位置(周期)が短時間方向に移動した(Fig. 3-1)。

②点滅周期を 1.0×10^{-3} s にした場合には、デューティ比を 100 ~ 10 % まで変化させても、光合成速度は、おおよそ一定だったが、点滅周期を 1.0×10^{-1} s にした場合には、デューティ比を下げるに従い光合成速度も低下した(Fig. 3-2)。

一方、デューティ比を下げると光合成が低下するという現象は、既報(森ら, 2002a)のLED点滅光を使用した測定結果でも発生し、この低下現象が発生するデューティ比は 10 % と報告されている。本章の実験でもデューティ比を 10 % にすると光合成速度が低下したため、この点では、既報(森ら, 2002a)と本章の実験結果の間には共通点があった(Fig. 3-2)。また、既報(森ら, 2002a)ではデューティ比を 33 % とした場合、連続光と比較してサラダナの成長率が約 25 % 増大したと報告されているが、本章の実験では、残念ながら既報の様に光合成速度が増大する現象は現れなかった(Fig. 3-2)。

この違いの要因は、恐らく、両者の実験で用いた試料の種類が、既報(森ら, 2002a)では「サラダナ」を用い、本章の実験では「ユーグレナ」を用いたため、これら試料の光合成機構の違いが関係したのではないかとと思われる。サラダナに照射するLED点滅光のデューティ比を 10 % とした場合は、点滅周期の明期が短いため、明期のPPFD値が $500 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 程度と強くなる(森ら, 2002a)。この $500 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ というPPFD値はサラダナの光飽和点を超えているので、この過剰な光量子によりサラダナがストレスを受けた可能性がある(森ら, 2002a)とのことだ。ここで、サラダナは高等植物なので光合成反応過程は「チラコイド反応(光化学系 I, II 等)」と「カルビン・ベンソン回路」と思われる。

「2.4.」で述べた様に、ユーグレナにも光化学系 I と II はあるので、基本的な光合成反応過程はサラダナなどの高等植物と共通である(横田, 1989)。しかし、「2.4.」で述べた様に、ユーグレナの光合成単位等は高等植物と比較して違いがある事が知られている。恐らく、これらの相違点が要因となり、サラダナとユーグレナでは光合成反応の「律速段階」が異なり、本章の結果(Fig. 3-2)が既報(森ら, 2002a)と異なる結果になったのではないかと推測する。

この実験では、LEDの種類を赤色LEDのみとした実験だったが、既報(松本ら, 2006,

2009)では赤色LEDの他に青色LEDも加えて、赤色LEDと青色LEDの2色を同時にユーグレナへ照射することで、ユーグレナの生育が良くなった報告された。従って、デューティ比や点滅周期の条件を変更しながら、赤色LEDと青色LEDの点滅光をユーグレナに同時照射すれば、さらなる光合成効率向上への糸口が得られると期待する。

3. 5 まとめ

この結果を産業上の利用面から考えると、例えば赤色LEDを用いてユーグレナ用の効率的な光源装置を設計する場合を考えると、実験結果から、LED照射光のPPFD設定値が $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 以下の条件であれば、

- ①「点滅周期は $1.0 \times 10^{-2} \text{ s}$ よりも短時間にすれば良い」ことと、PPFD設定値を光合成効率が高い「約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 」の条件とした場合、
- ②「デューティ比の設定範囲は、10 ~ 100 % にすれば良い」ことが分かった。

これらの結果は今後、「ユーグレナ用植物工場」を開発する上で非常に有益な情報と思われる。

一方、本章では赤色LEDのみを用いて実験したため、既報(松本ら, 2006, 2009)の様に青色LED光と赤色LED光を同時に照射した場合の光合成速度の増加現象については分からなかった。つぎの実験では、青色LEDと赤色LEDの点滅光を同時にユーグレナへ照射することで、より高効率な光合成条件を目指す。

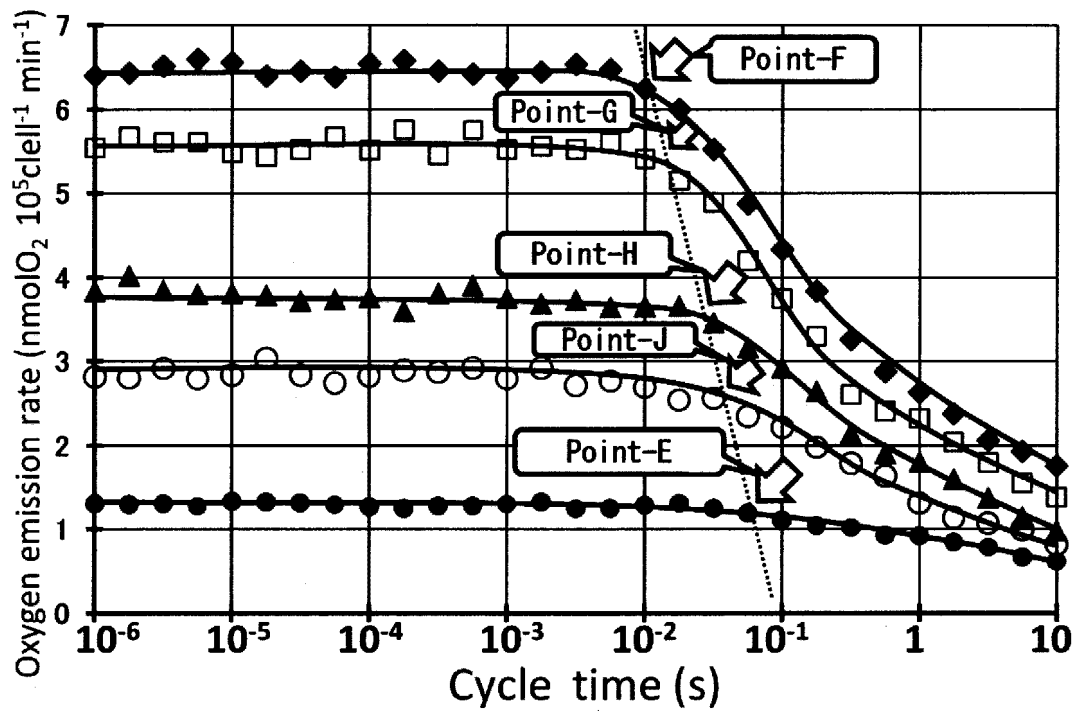


Fig. 3 - 1 Effects of the cycle time of blinking red LED lights on oxygen emission rates under several PPFD values for duty ratio of 50 %.

- ◆: PPFD = 400 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$
- : PPFD = 218 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$
- ▲: PPFD = 118 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$
- : PPFD = 64 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$
- : PPFD = 35 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$

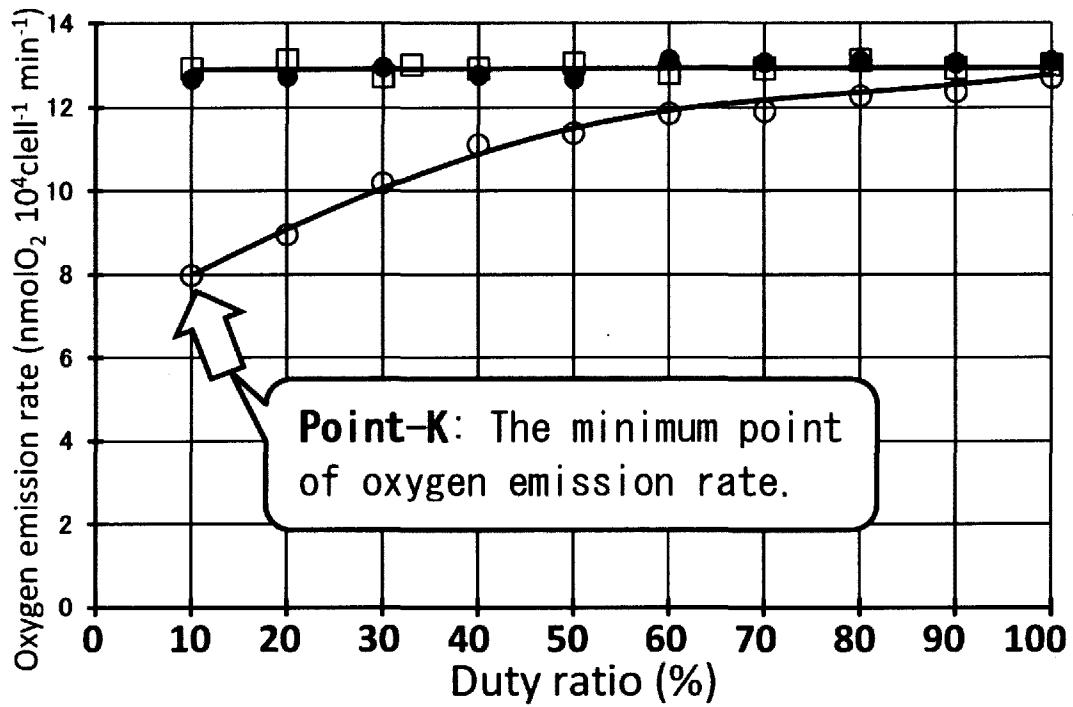


Fig. 3 - 2 Effects of the duty ratio of blinking red LED lights on oxygen emission rates for PPFD of $35 \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$.

- : Cycle time = 4.0×10^{-4} s
- : Cycle time = 1.0×10^{-3} s
- : Cycle time = 1.0×10^{-1} s

第4章

青色と赤色LED点滅光の点滅周期と
デューティ比がユーグレナの
光合成速度に及ぼす影響

4. 1. 序論

第 3 章にて、光量を段階的に変化させた場合の「光合成速度に対する点滅周期の影響」や「光合成速度に対する点滅デューティ比の影響」の測定を行った。その結果、ユーグレナに適した点滅周期やデューティ比を明らかにした。しかし、第 3 章では、光源に赤色LEDのみを使用し、赤色光照射の条件下のみで光合成速度を測定したため、照射光の波長による影響は、分からなかった。

照射光の波長による影響に関して、既報ではユーグレナに対して 2 種類の波長を同時に照射すると光合成速度が向上したと報告された(松本 ら, 2006, 2009)。具体的には、ユーグレナに対して青色LEDと赤色LEDの連続光をR/B比(赤と青の光量子数の比率) = 9 の混合照射条件下で照射することで、光合成速度が向上したと報告された。さらに、サラダナに対して、LED点滅光の点滅周期を 4.0×10^{-4} s の条件で照射すると、連続光と比較して光合成速度が約 23 % 増大したと報告された(森 ら, 2002a)。従って、筆者らは、2 波長の混合照射や点滅周期の相乗効果を期待して、ユーグレナに対して青色光と赤色光の 2 種類の波長を同時に照射した条件下における「光合成速度に対する点滅周期の影響」を測定した。

さらに、既報ではサラダナに対してLED点滅光のデューティ比を 33 % の条件で照射すると、連続光と比較して光合成速度が 25 % 高くなったと報告された(森 ら, 2002a)。もしかすると、ユーグレナに対して 2 波長の混合照射条件下で、さらにデューティ比を最適化することで光合成速度が向上するかもしれない。従って、つぎに筆者らは、ユーグレナに対して青色光と赤色光の 2 種類の波長を同時に照射した条件下における「光合成速度に対する点滅デューティ比の影響」を調べた。本実験の主目的は、上述した既報での混合照射条件や点滅周期条件、点滅デューティ比条件の良好な光質条件をユーグレナに対して同時に適用することで、相乗効果等のため、さらなる光合成効率向上を図ることである。調査内容の詳細を以下に報告する。

4. 2. 材料及び実験方法

4. 2. 1. 試料と培養方法

測定用試料には、ユーグレナ(*Euglena gracilis* Z 株)を培養して用いた。培養方法は、「2.2.1.」に従った。

4. 2. 2. 試料の調整方法

試料の調整方法や、測定チャンバーへ試料を注入する方法や、測定準備の手順は、「2.2.2.」に従った。

4. 2. 3. LED光源の仕様、及び構成

実験に用いる光源として、下記仕様のLED光源ユニットを製作した。

①照射光の中心波長(λ): 625 nm(赤)及び、470 nm(青) (分光特性: Fig. 4-1)

ここで用いた赤LED($\lambda = 625$ nm)は、2 章や 3 章で用いた赤LED($\lambda = 660$ nm)と比較して波長が若干違うので以降、波長 = 625 nmのLEDを「赤LED2」と記述する。

②青LED (OSUB5161P, OptoSupply Ltd) と赤LED2 (OSHR5161P, OptoSupply Ltd) を照明用のLEDプリント基板に合計 540 個実装して用いた。光を均一に混合する目的で青LEDと赤LED2を図の様に交互に配置した (Fig. 4-2)。

「2.2.3.2」と同様に、プリント基板の中央部に 120 ピンのカードエッジコネクタを配置した。このコネクタに「2.2.3.3」で記載した「LEDドライブ基板」を接続することで、各LEDに電流を供給するという構造である。プリント基板上での各LEDの配線は、まず 1 列目に並んだ 9 個のLEDを直列に接続し、両端のプラス側端子とマイナス側端子をカードエッジコネクタの異なる端子へ接続した。同様に残りの各LEDも列毎にカードエッジコネクタの異なる端子へ接続し、全 60 列の直列回路をカードエッジコネクタへ接続した。この様に、9 個のLEDを直列接続した列を 60 列作り各端子をカードエッジコネクタへ接続することで、全540 個のLEDに電流を供給

する回路を製作した（回路図：Fig. 4-3）。

さらに、光を均一に拡散する目的でLEDの前面に光拡散板(LSD80PC10-F12, Optical solutions Ltd.)を配置した。

③PPFD設定値と点滅周期と点滅デューティ比と点滅動作の制御方法は、「2.2.3.」に従った。

4. 2. 4. 光合成速度の測定方法

「2.2.4.」に従い、クラーク電極タイプの液相酸素濃度測定器(OXY-1, Hansatech Instruments Ltd)とLED光源ユニットを組合せて測定システムを構築した(Fig. 4-4)。LED光源ユニットの点滅点灯の制御方法や液相酸素濃度の測定手順や光合成速度の算出方法やPPFD値の設定方法や周囲環境条件は、「3.2.4.」に従った。

4. 2. 5. 実験方法

4. 2. 5. 1. 光合成速度に対する点滅周期の影響を測定

測定に用いた「点滅光のタイミング波形」や「PPFD設定値」の調整方法は、「2.2.6.1.」に従った。まず初めに、青色と赤色LED2を用いて、PPFD設定値 = $30 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 、点滅光のR/B比 = 9 (赤LED2 = $27 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 、青LED = $3 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 、簡便のため以降「R/B比 = 9」での照射を「青赤混合照射」と記載する)、点滅デューティ比 = 33% の条件に固定した。

この条件下で、点滅光の周期を変化させながら、青色と赤色の点滅光を同時にユーグレナに照射して、光合成速度を測定した。さらに、コントロールとして、赤色LED2による「赤色単色光での照射」（簡便のため以降、「赤単色照射」と記載する）条件についても同じ条件にて測定した。

4. 2. 5. 2. デューティ比の影響を測定

測定に用いた「点滅光のタイミング波形」やPPFD設定値の調整方法は、「2.2.6.1.」に従った。青赤混合照射でのPPFD設定値を $30 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ とし、点滅周期を $4.0 \times 10^{-4} \text{ s}$ の条件に固定した。この固定条件下で、点滅デューティ比を変化させながら、光合成速度を測定した。

4. 3. 結果

4. 3. 1. 光合成速度に対する点滅周期の影響

点滅周期を $1.0 \times 10^{-6} \sim$ 約 $5.0 \times 10^{-3} \text{ s}$ とした場合、光合成速度は約 $12 \text{ nmol O}_2\cdot 10^4\text{cell}^{-1}\cdot\text{ml}^{-1}\cdot\text{min}^{-1}$ の値となり、おおよそ一定の値だった (Fig. 4-5)。点滅周期を約 $5.0 \times 10^{-3} \text{ s}$ (Point-L) よりも長時間にするに伴い、光合成速度は、低下した。特に、点滅周期を $1.0 \times 10^{-1} \sim 1 \text{ s}$ の範囲で変化させた場合には、図中の矢印で示した様に光合成速度は急激に低下した (Fig. 4-5)。

対照実験の「赤単色照射」条件下の結果と比較すると、点滅周期を $1.0 \times 10^{-6} \sim 3.0 \times 10^{-1} \text{ s}$ (Point-M) とした周期領域では、両者の曲線は重なっており、おおよそ同じ光合成速度だった。しかし、点滅周期を Point-M よりも長時間に設定すると、「青赤混合照射」条件下と比較して、「赤単色照射」条件下での光合成速度は、急激に低下した。

また、図中の Point-N は、既報(森ら, 2002a)にて光合成速度が増大した、と報告された「点滅周期 = $4.0 \times 10^{-4} \text{ s}$ 」の条件の点である。この点での光合成速度は、これより短い周期条件とおおよそ同等だった (Fig. 4-5)。

4. 3. 2. デューティ比の影響

点滅デューティ比を 10 ~ 100% に変化させた場合、光合成速度は、約 $12 \text{ nmol O}_2\cdot 10^4\text{cell}^{-1}\cdot\text{ml}^{-1}\cdot\text{min}^{-1}$ の値となり、おおよそ一定の値だった (Fig. 4-6)。また、図中の

Point-P は、既報(森ら, 2002a)にてサラダナの成長率が高くなったと報告された「点滅デューティ比 = 33 %」の条件の点である。しかし、この点での光合成速度は、他の点滅デューティ比条件とおおよそ同等だった (Fig. 4-6)。

4. 4. 考察

本章の目的である、『既報にて報告された良好な光質条件を同時に適用した光質を用いて、青赤混合照射下における「光合成速度に対する点滅周期の影響」と「光合成速度に対する点滅デューティ比の影響」に関するデータを得ること』が出来た。得られたデータの特長をまとめると、

①連続光と比較して、光合成速度が高くなる点滅周期やデューティ比の条件は、なかった (Fig. 4-5, 4-6)。

②点滅周期を 1.0×10^{-6} ~ 約 5.0×10^{-3} s とした場合、光合成速度は、おおよそ一定の値だったが、点滅周期を約 5.0×10^{-3} s (Point-L) よりも長時間にすると、光合成速度は、低下した (Fig. 4-5)。補足だが、結果「2.3.3.」での変化点 (周期) と比較して、ここでの変化点の周期はより短時間になっている。この理由は「2.3.3.」ではデューティ比を 50 % としたが、ここではデューティ比を 33 % としたため、デューティ比が異なるため光合成曲線が変化したものと思われる (Fig. 4-5)。

③点滅デューティ比を著しく下げ、10 % にしても光合成速度は、他の点滅デューティ比条件と同等だった (Fig. 4-6)。

既報(森ら, 2002a)では、点滅周期を 4.0×10^{-4} s とした場合に、連続光と比較してサラダナの光合成速度が約 23 % 増大し、さらに点滅デューティ比を 33 % にすると、サラダナの成長率が 25 % 高くなったと報告された。しかし、本章の実験では、既報(森ら, 2002a)の様に、光合成速度が増大する現象は発生しなかった (Fig. 4-5, 4-6)。

この違いの要因は、恐らく、両者の実験で用いた試料の種類が、既報(森ら, 2002a)では「サラダナ」を用い、本章の実験では「ユーグレナ」を用いたため、これら試料の

光合成機構の違いが関係したのではないかとと思われる。

「3.4.」で述べた様に、サラダナに照射するLED点滅光のデューティ比を10%とした場合は、明期のPPFD値がサラダナの光飽和点を超え、サラダナがストレスを受けた可能性があることが要因と思われる。さらに、「2.4.」で述べた様に、ユーグレナの光合成単位等は高等植物と比較して違いがある事も知られている。さらに、サラダナのような高等植物では、フォトトロピン等の青色光受容体が気孔開口反応に関与している(和田ら, 2001)が、ユーグレナに気孔はない。恐らく、これらの相違点が要因となり、サラダナとユーグレナでは光合成反応の「律速段階」が異なり、本章の結果(Fig. 4-5, 4-6)が既報(森ら, 2002a)と異なる結果になったのではないかと推測する。

また、「青赤混合照射」と「赤単色照射」との結果を比較すると、点滅周期を約 3.0×10^{-1} s(Point-M)よりも長くした条件下では、「赤単色照射」の方が光合成速度は低下した。(Fig. 4-5) 特に、点滅周期 = 10 s の条件下での両者を比較すると、「赤単色照射」にした場合の光合成速度は、「青赤混合照射」にした場合の光合成速度の約半分にまで低下した。

この光合成速度の低下現象が発生した理由として以下の推論が考えられる。

- ①光合成に利用される光量子の大部分は、チラコイド膜上のLHC(Light-harvesting complex)にて補足される(山崎, 2011)。
- ②LHCにて、光量子は、クロロフィルaやクロロフィルbやカロテノイド等の色素に補足される。(Lincoln・Eduardo, 2004; 山崎, 2011)
- ③青色光は、これら全ての色素に補足され易いが、赤色光は、波長の違いが原因で、カロテノイドに補足され難い性質がある(山崎, 2011)。
- ④従って、「赤単色照射」よりも、「青赤混合照射」の方が、カロテノイドに青色光が補足され易いため、より多くの光量子が光合成に利用される可能性があると思われる。
- ⑤以上の理由で、「青赤混合照射」と比較して、「赤単色照射」の場合の光合成速度が低下するという「低下現象」が発生したと思われる。
- ⑥点滅周期を Point-M よりも長時間にした条件下のみで、上述の「低下現象」が発生

した理由は、恐らく、点滅周期を Point-M よりも長時間にしたことで、光合成の律速段階がLHCと深く関係する部分へと変化したのではないかと思われる。しかし、本章の実験結果だけで、上述の推論を断定することは難しいと思われる。

つぎに、光合成の「光エネルギー利用効率」の観点から考察する。「赤単色照射」(単一波長)条件の場合、光合成の「光エネルギー利用効率」が最大になる光質条件下では、「光合成PPFD効率」も最大になると考えられる。なぜならば、照射光が単一波長ならば、光量子エネルギーが全光量子で同じなので照射した光エネルギーと光量子数は比例関係になるからである。従って「赤単色照射」の場合、「光合成PPFD効率」を最大にする光質条件下では、光合成の「光エネルギー利用効率」も最大になる。

しかし、「青赤混合照射」の場合には、青色光の波長が赤色光と異なるため、青色光の光量子エネルギーと赤色光の光量子エネルギーは異なる。すなわち、「赤単色照射」の光量子数と「青赤混合照射」の光量子数を同じ光量子数にしても、両者の総光量子エネルギーは異なる。従って、「青赤混合照射」と「赤単色照射」の「PPFD値」を同じ値に合わせて光合成速度を単純に比較しただけでは、両者の「光エネルギー利用効率」を比較することは出来ない。

そこで「光エネルギー利用効率」を算出するには、光量子エネルギーを考慮した、以下の計算が必要である。まず、赤LED2の光量子エネルギーを計算する。光量子エネルギーは、以下の式(4-1)で定義されている (Lincoln・Eduardo, 2004) ので、これに赤LED2の波長 ($\lambda = 625 \text{ nm}$) を代入して光量子のエネルギーをE(led2)として式(4-2) の様に計算した。

$$E = h \times c / \lambda \quad (\text{式4-1})$$

- ・ E : 光量子エネルギー (J)
- ・ h : プランク定数 = 6.626×10^{-34} (Js)
- ・ c : 光速 = 3.0×10^8 (ms^{-1})
- ・ λ : 光の波長 (m)

$$\begin{aligned}
E(\text{led2}) &= h \times c / \lambda (\text{led2}) \quad (\text{式4-2}) \\
&= 6.626 \times 10^{-34} \times 3.0 \times 10^8 / 625 \text{ nm} \\
&= 3.180 \times 10^{-19} (\text{J})
\end{aligned}$$

つぎに、赤LED2の光量をPPFD設定値 = $30 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ に設定して照射した場合の、単位面積当たりの光エネルギーを求める。単位面積当たりの光エネルギーは、「放射照度(Irradiance)」と定義されている(長倉ら, 1998)。この定義から単位面積当たりに照射された全光量子エネルギーは、放射照度(I)と等しいので、式(4-3)を得る。

$$I(\text{W}\cdot\text{m}^{-2}) = E(\text{J}) \times \text{PPFD設定値} (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \quad (\text{式4-3})$$

従って、式(4-3)に式(4-2)の値とPPFD設定値を代入して計算すると、赤LED2の光量をPPFD設定値 = $30 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ に設定して照射した場合の、単位面積当たりの光エネルギーになる。この光エネルギーをI(r30)として式(4-4)の様に計算した。

$$\begin{aligned}
I(\text{r30}) &= E(\text{led2}) \times \text{PPFD}(30) \quad (\text{式4-4}) \\
&= 3.18 \times 10^{-19} (\text{J}) \times 30 (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \\
&= 3.18 \times 10^{-19} \times 30 \times 10^{-6} \times N (\text{mol}^{-1}) \\
&= 3.18 \times 10^{-19} \times 30 \times 10^{-6} \times 6.022 \times 10^{23} \\
&= 5.745 (\text{W}\cdot\text{m}^{-2})
\end{aligned}$$

$$\cdot N: \text{アボガドロ定数} = 6.022 \times 10^{23} (\text{mol}^{-1})$$

但し、各LEDの波長分布をFig. 4-1 で観察すると厳密には波長帯域幅があり、単一波長のみでなく複数の波長を含むので正確なエネルギー値を算出するためには、全波長

領域で積分計算をする必要がある。しかし、各LEDの帯域幅が非常に狭いので、ここでは簡便のためLEDの波長を単一波長として計算した。

つぎに、「4.2.5.1.」で記載した様に「青赤混合照射」時の赤LED2のPPFD設定値は $27 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ であり、青色光のPPFD設定値は $3 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ なので、各色毎にエネルギー値を算出する。式(4-4)と同様に赤LED2をPPFD設定値 = $27 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の値で照射した場合の放射照度を $I(r27)$ として式(4-5)の様に計算した。

$$\begin{aligned}
 I(r27) &= E(\text{led2}) \times \text{PPFD}(27) \quad (\text{式4-5}) \\
 &= 3.18 \times 10^{-19}(\text{J}) \times 27 (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \\
 &= 3.18 \times 10^{-19} \times 27 \times 10^{-6} \times N (\text{mol}^{-1}) \\
 &= 3.18 \times 10^{-19} \times 27 \times 10^{-6} \times 6.022 \times 10^{23} \\
 &= 5.170 (\text{W}\cdot\text{m}^{-2})
 \end{aligned}$$

式(4-2)と同様に、青色光 ($\lambda = 470 \text{ nm}$) の光量子エネルギーを $E(\text{blue})$ として式(4-6)の様に計算した。

$$\begin{aligned}
 E(\text{blue}) &= h \times c / \lambda (\text{blue}) \quad (\text{式4-6}) \\
 &= 6.626 \times 10^{-34} \times 3.0 \times 10^8 / 470 \text{ nm} \\
 &= 4.229 \times 10^{-19} (\text{J})
 \end{aligned}$$

さらに、青色光をPPFD設定値 = $3 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ に設定して照射した場合の放射照度を $I(b3)$ として式(4-7)の様に計算した。

$$\begin{aligned}
 I(b3) &= E(\text{blue}) \times \text{PPFD}(3) \quad (\text{式4-7}) \\
 &= 4.229 \times 10^{-19}(\text{J}) \times 3 (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \\
 &= 4.229 \times 10^{-19} \times 3 \times 10^{-6} \times N (\text{mol}^{-1})
 \end{aligned}$$

$$\begin{aligned}
&= 4.229 \times 10^{-19} \times 3 \times 10^{-6} \times 6.022 \times 10^{23} \\
&= 0.764 \text{ (W}\cdot\text{m}^{-2}\text{)}
\end{aligned}$$

各色毎のエネルギー値を算出したので、「青赤混合照射」時の総光エネルギーを $I(r\&b)$ として式(4-8)の様に計算した。

$$\begin{aligned}
I(r\&b) &= I(r27) + I(b3) \quad (\text{式4-8}) \\
&= 5.170 \text{ (W}\cdot\text{m}^{-2}\text{)} + 0.764 \text{ (W}\cdot\text{m}^{-2}\text{)} \\
&= 5.934 \text{ (W}\cdot\text{m}^{-2}\text{)}
\end{aligned}$$

この値と「赤単色照射」時の放射照度 $I(r30)$ とを比較する目的で、両者の比を $RI(r\&b/r30)$ として式(4-9)の様に計算した。

$$\begin{aligned}
RI(r\&b/r30) &= I(r\&b) / I(r30) \quad (\text{式4-9}) \\
&= 5.934 \text{ (W}\cdot\text{m}^{-2}\text{)} / 5.745 \text{ (W}\cdot\text{m}^{-2}\text{)} \\
&= 1.033
\end{aligned}$$

この結果から、「赤単色照射」と「青赤混合照射」のPPFD設定値を同じ値の $30 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ に設定しても、「青赤混合照射」時の光エネルギーは、「赤単色照射」時の光エネルギーよりも約 1.033 倍、大きいと言える。言い換えると、「赤単色照射」の方が、光エネルギーは少ない。

以上の結果を踏まえて、Fig. 4-5 を観察すると、Point-M よりも短時間の周期領域では、「赤単色照射」の曲線と「青赤混合照射」の曲線が重なっており、おおよそ同じ光合成速度である。特に Point-N 付近の光合成曲線が平坦な周期領域は、曲線が最も高い値になるため、光合成速度も最高値になる周期領域である。以後簡便のため、この周期領域を「最適周期領域」と記載する。

さらに、式(4-9)の結果から、同じPPFD設定値の条件下では、「赤単色照射」の方が光エネルギーが少ないことも分かっている。従って「最適周期領域」では、「青赤混合照射」よりも「赤単色照射」の方が「光エネルギー利用率」が高い可能性があると思われる。

また、視点を変えて、青色LEDと赤色LEDにおける「電気エネルギー」から「光エネルギー」への変換効率に着眼すると、単位時間当たり同じ光量子数で発光させた場合、青色LEDは赤色LEDの約2倍の電気エネルギーを消費する（高辻・森，2011）。すなわち、同じPPFD値で両者を発光させた場合、赤色LEDは青色LEDの約半分の電気エネルギーで済むことになるため、上述の結果とも合致する。

4. 5. まとめ

本章の結果を産業上の利用面から考えると、上述の「最適周期領域」の条件で、「ユーグレナ用の効率的な照明装置」を設計する場合、本章の実験結果から、点滅デューティ比が 33 % の場合には、「LEDの点滅周期を約 5.0×10^{-3} s 以下」とすれば、光合成PPFD効率が低下しないことが分かった。

また、青赤混合照射条件下にて、点滅周期を 4.0×10^{-4} s に設定した場合には、点滅デューティ比を 10 ~ 100 % の範囲で変化させても、光合成効率が低下しないことも分かった (Fig. 4-6)。これらの結果は今後、「ユーグレナ用植物工場」を開発する上で非常に有益な情報と思われる。

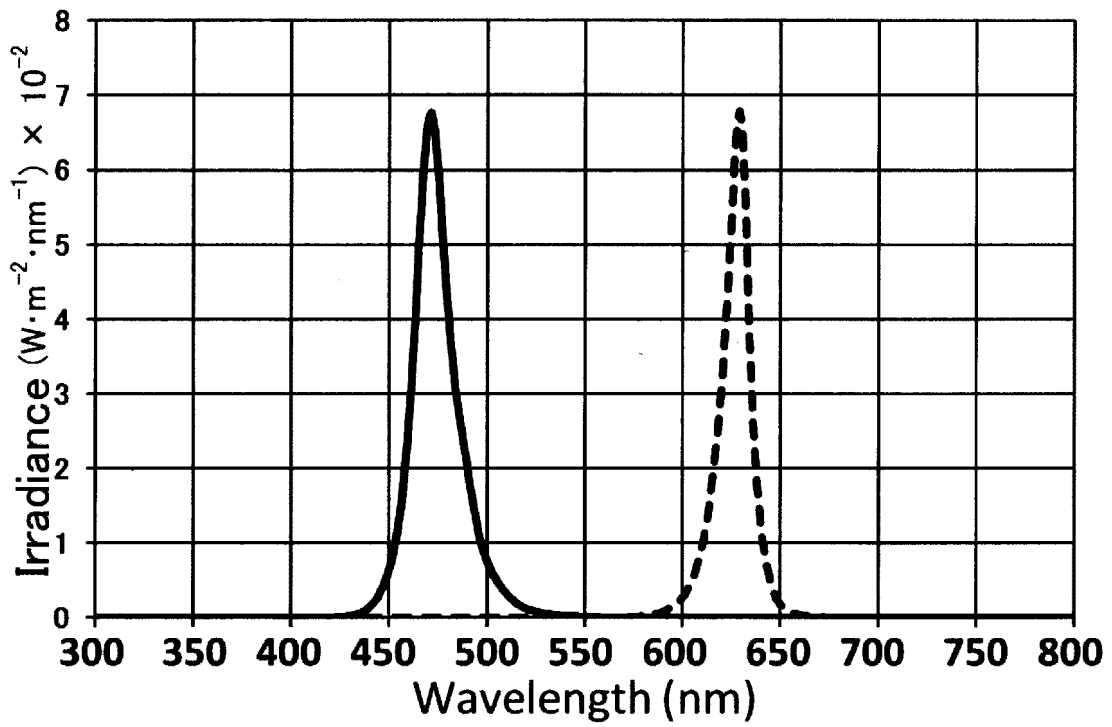


Fig. 4 - 1 Spectral distribution of blue and red LEDs.
The dashed and solid lines represent the red and blue LEDs, respectively.

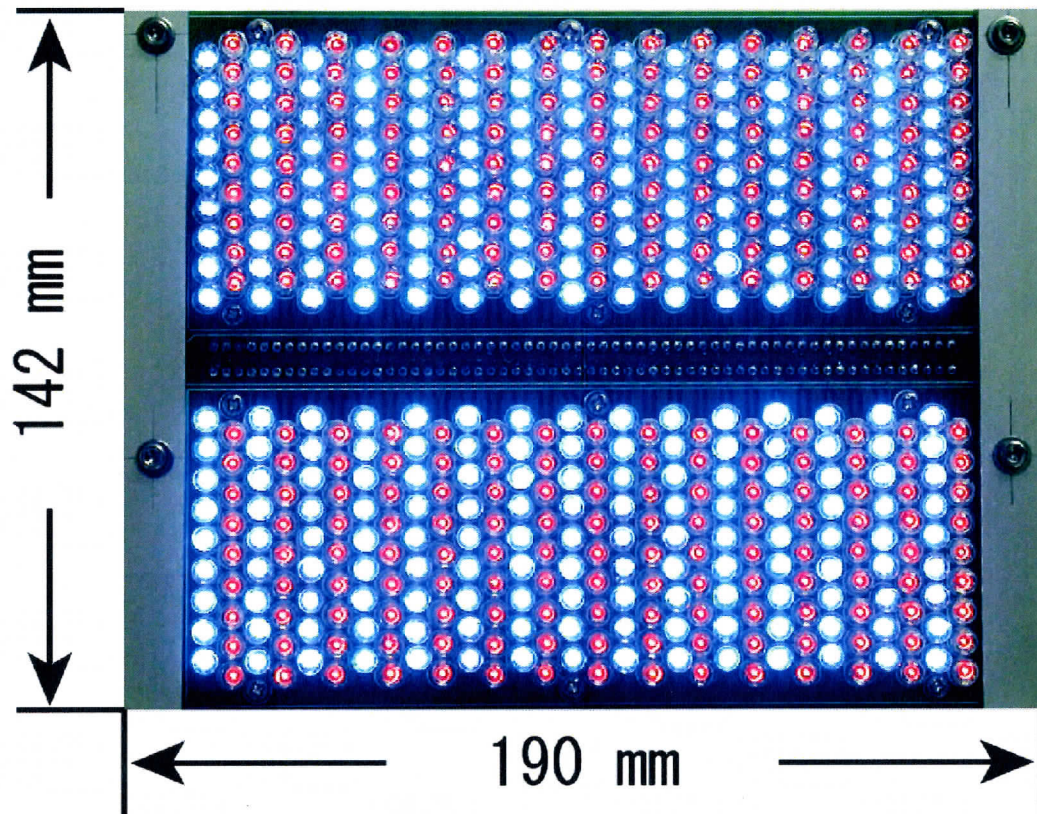


Fig. 4 - 2 The Blue and red2 LED panel.

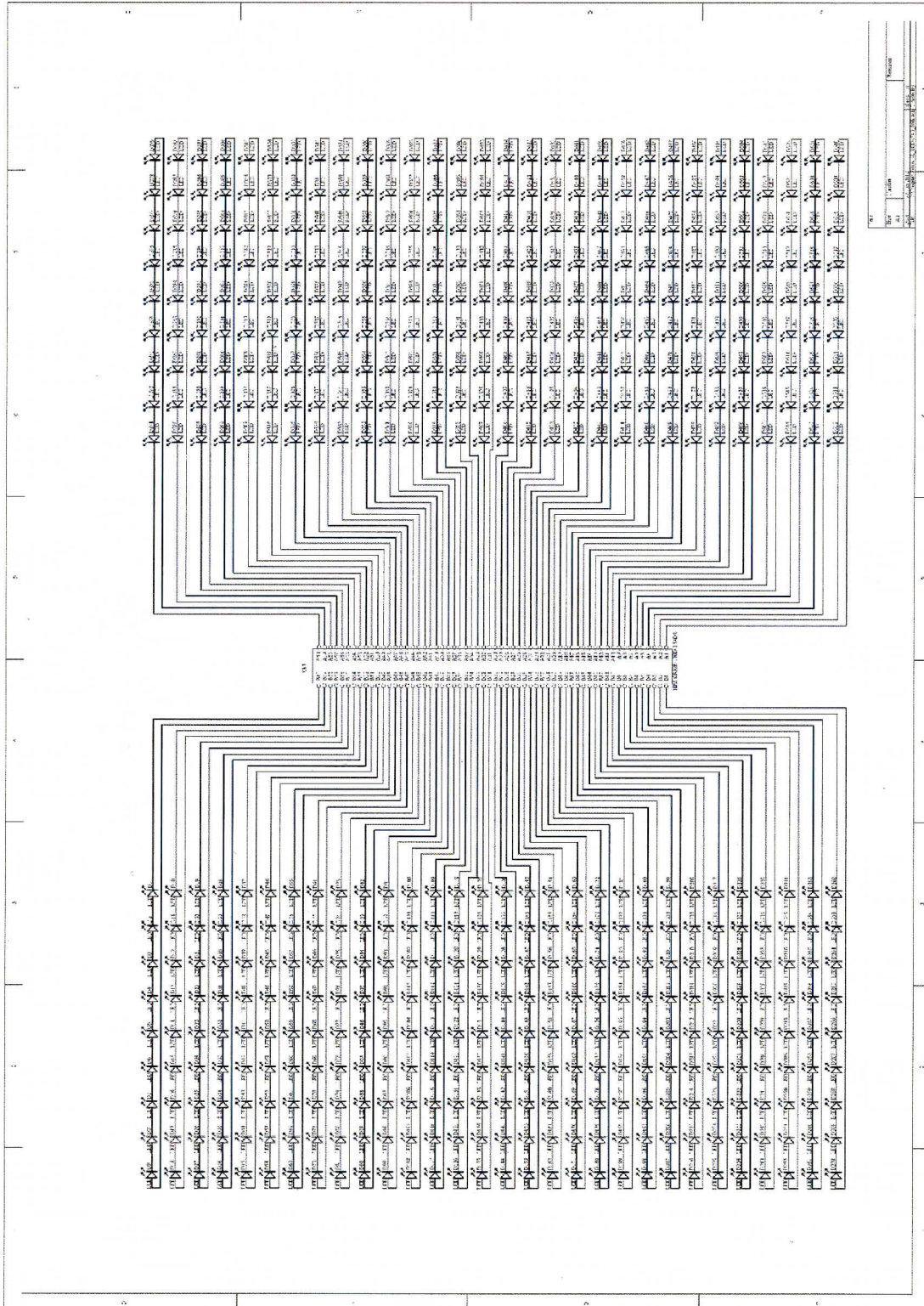


Fig. 4 - 3 The circuit diagram of LED2 printed-circuit board.

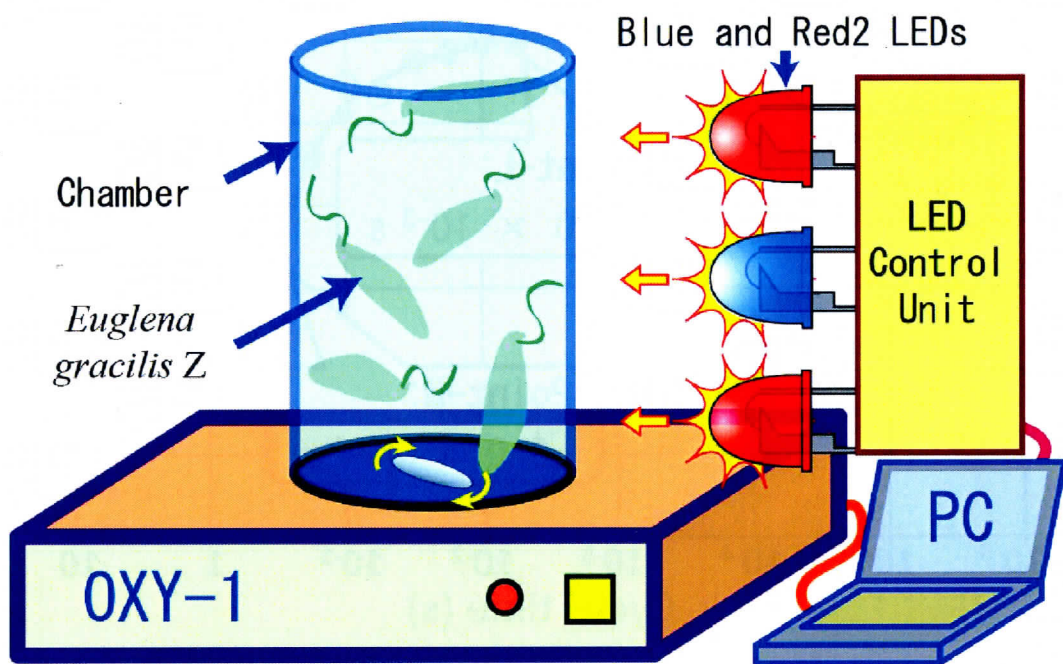


Fig. 4 - 4 The equipment used to measure the photosynthesis rate.

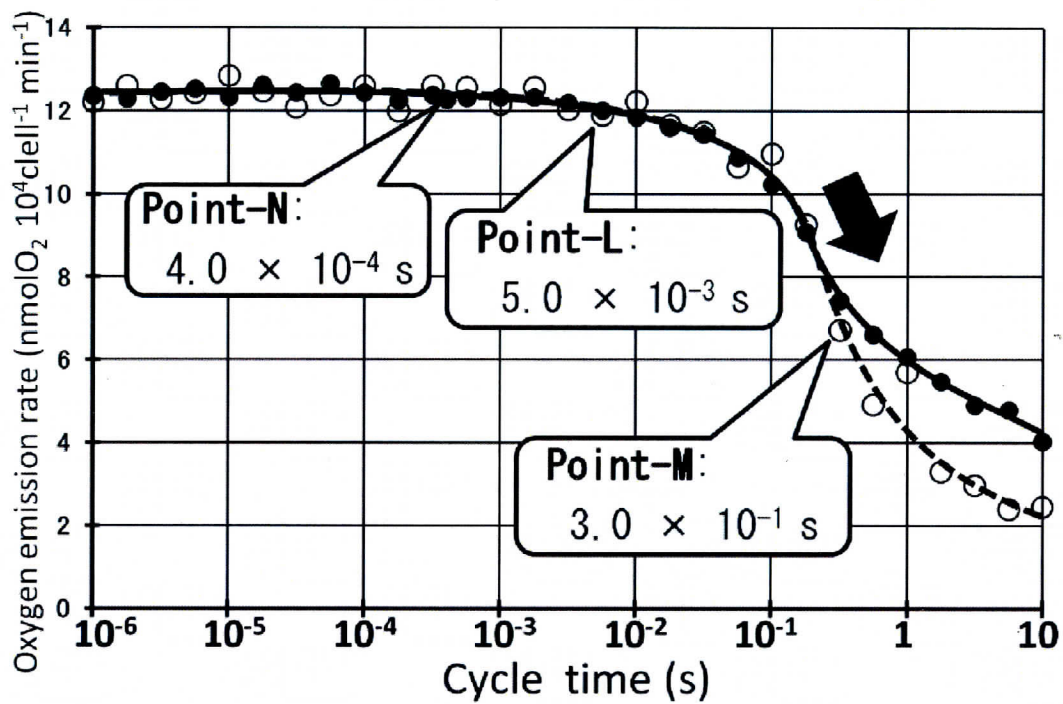


Fig. 4 - 5 Effects of the cycle time of blinking blue and red2 LED lights on oxygen emission rate for PFD = $30 \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$, and duty ratio = 33 %.

●: Blue and red2 (R / B = 9)

○: Red2 only

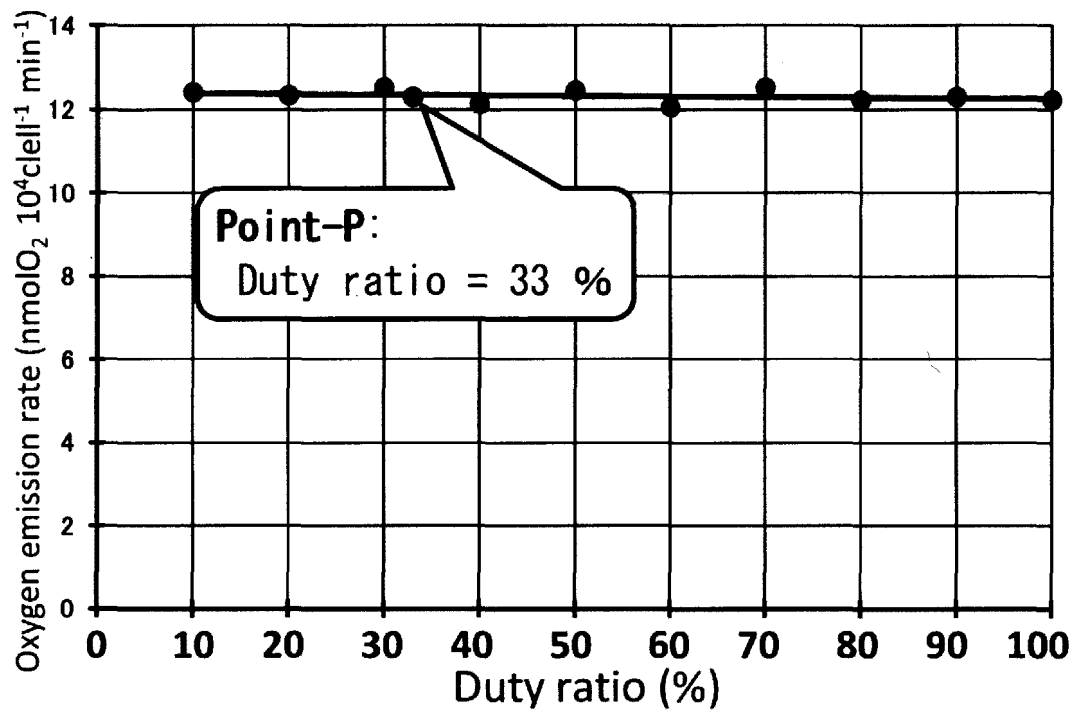


Fig. 4 - 6 Effects of the duty ratio of blinking blue and red2 LED lights on oxygen emission rates for PPF_D = 30 μmol · m⁻² · s⁻¹, R/B = 9, and cycle time = 4.0 × 10⁻⁴ s .

第5章

総合考察

本研究の目的は、ユーグレナに対し様々な光質条件のLED光を照射し「光-光合成曲線」などのデータを測定することで、効率的な照射条件を明確化し、より高い収量を得られる栽培条件を明らかにすることである。すなわち、光質条件のPPFD値や点滅周期、点滅デューティ比、光の波長に着眼し、光合成速度の指標として酸素発生速度に基づき評価を行い、ユーグレナにとって最適な光質条件の明確化を目指した。

ユーグレナは、栄養価が非常に高くさらに光合成効率も高いため新たな食資源としてやバイオ燃料（佐伯ら, 2008; 出雲, 2011）としても注目されており、食糧問題やエネルギー問題の解決手段として期待されている。ユーグレナの大量培養は、屋外での培養事例（佐伯ら, 2008）があるものの、①日照や天候の影響を受けること、②雑菌が混入する可能性があること、③病虫害や異物が風で混入する可能性がある、などの問題があり収量や品質が安定しない恐れがある。

さらに、屋外で培養する場合の照射光は、太陽光である。地表における太陽光のPPFD値の最大値は、約 $2000 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ である（柳・岡本, 2004）。また、第2章の図（Fig. 2-15）からユーグレナの光飽和点は約 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ である。従って、太陽光の $2000 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ という光量は、ユーグレナの光飽和点（約 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ ）の約5倍という非常に強い光量である。ここで、太陽光と同じ $2000 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ の光量を照射した場合のユーグレナにおける「光合成PPFD効率」を「 η_{sun} 」として概算する。第2章の図（Fig. 2-15）からPPFD値を約 $1600 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ とした場合の酸素発生速度は約 $6.5 \text{ nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1}$ である。さらに、光飽和点以上のPPFD値では曲線がほぼ平坦になっているため、PPFD値を $2000 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ にしてもほぼ同じ酸素発生速度になると仮定する。つぎに、これらの値を「2.2.5.」に記載した「光合成PPFD効率」の定義式（式2-2）へ代入し「 η_{sun} 」を算出する。

$$\begin{aligned}\eta_{\text{sun}} &= \text{酸素発生速度} (\text{nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1}) / \text{PPFD値} (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \\ &= 6.5 (\text{nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1}) / 2000 (\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \\ &= 3.25 \times 10^{-3} (\text{nmolO}_2\cdot\text{ml}^{-1}\cdot\text{min}^{-1} / \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}) \quad (\text{式5-1})\end{aligned}$$

つぎに、第 2 章の図 (Fig. 2-16) で得られた、光合成PPFD効率を最大にする点 (Point-C) での「光合成PPFD効率」を「 η_{\max} 」として概算する。図 (Fig. 2-16) から光合成PPFD効率が最大になる点 (Point-C) の座標値は、酸素発生速度が約 $2.5 \text{ nmolO}_2 \cdot \text{ml}^{-1} \cdot \text{min}^{-1}$ であり、PPFD値は約 $35 \text{ } \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$ である。従って、同様に式2-2 へ代入し算出する。

$$\begin{aligned} \eta_{\max} &= \text{酸素発生速度 (nmolO}_2 \cdot \text{ml}^{-1} \cdot \text{min}^{-1}) / \text{PPFD値 (}\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}) \\ &= 2.5 \text{ (nmolO}_2 \cdot \text{ml}^{-1} \cdot \text{min}^{-1}) / 35 \text{ (}\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}) \\ &= 7.14 \times 10^{-2} \text{ (nmolO}_2 \cdot \text{ml}^{-1} \cdot \text{min}^{-1} / \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}) \quad (\text{式5-2}) \end{aligned}$$

従って、両者を比較するため、太陽光と同じ光量での光合成PPFD効率と、Point-C ($35 \text{ } \mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$) での光合成PPFD効率の比率を $R(\max/\text{sun})$ として算出する。

$$\begin{aligned} R(\max/\text{sun}) &= \eta_{\max} / \eta_{\text{sun}} \\ &= 7.14 \times 10^{-2} / 3.25 \times 10^{-3} \\ &= 21.96 \\ &= \text{約 22 倍} \quad (\text{式5-3}) \end{aligned}$$

この結果から、第 2 章で得られた、光合成PPFD効率を最大にする点 (Point-C) での光量条件で照射した場合と、太陽光の光量条件で照射した場合のユーグレナにおける光合成PPFD効率を比較すると、Point-Cの光量条件の方が太陽光の場合よりも、約 22 倍も光合成PPFD効率が高くなることが分かった。すなわち、太陽光を直接照射して培養するよりも、光量を Point-C の PPFD値に制御して照射することで、約 22 倍も効率が向上するということである。実際には、太陽光の光量は天候や季節等で変動するので誤差はある。しかし、両者の光合成PPFD効率には、極めて大きな違いがあると言える。

この結果を視点を変えて考えると、例えば、太陽光を「シリンドリカルレンズと導光板で培養槽へ照射する方式」(宮武, 1999) を応用して、Point-C の光量値に分割し

適切に培養槽へ照射することで、光合成PPFD効率を約 22 倍も向上させることが出来る
と言える。

また、太陽光の場合には、昼と夜があるため夜の期間は暗くなる。従って、夜間
(暗期)には光がないため光合成を出来ない。さらに、暗期には、生命活動維持に必
要な呼吸が行われるため、蓄積された糖質などの光合成産物が消費され減少する。ユ
ーグレナは夜(暗期)がなくとも良好に育つことが知られている(松本ら, 2006)。
従って、夜間に人工光を照射し24時間光合成を行うことで、暗期での光合成産物の減少
をなくし、さらなる効率的な培養が可能と思われる。

上述の様に、実際の培養では、太陽光に導光板を応用する方式や人工光を併用する
方式、太陽光を利用せずに人工光のみを照射する方式など多様な照射方法が考えられる
が、光量設定値の違いにより「光合成PPFD効率」が約 22 倍も異なることは、特筆すべ
き重要な照明設計の技術的要素である。

一方、近年日照や気温などの気候条件に影響されず安定的に植物を栽培するシステ
ムとして、「植物工場」と呼ばれる照明や空調等を用いた環境制御型の施設栽培システ
ムが実用化されている(高辻, 2006)。植物工場の特長は病害虫がいない清潔な環境中
で土壌を使わない水耕栽培が主流であり、無農薬で安全・安心な作物を天候に左右され
ず安定に周年生産可能なことである(高辻, 2006)。さらに植物工場は都会のビルや地
下空間にも設置可能なため消費地間近で生産可能であり、また各栽培槽を積み重ねた多
段栽培により土地の利用効率向上も可能になる(高辻, 2006)。

ユーグレナは培養液中での高密度培養が可能であり、培養液の温度や組成、光条件
等を適正化すれば閉鎖系で工業的に培養可能(宮武, 1999)であるため、植物工場の技
術を応用し、高度な環境制御システムを適用した「ユーグレナ用植物工場」を実現すれ
ば、狭い土地で安全・安心な高栄養価の食資源を大量生産し安価に安定供給することが
可能になると思われるが、今はまだ実現していない。

「ユーグレナ用植物工場」を実現する上で重要な課題の一つは「光環境の最適化」
であり「最適な照明システム」だと思われる。そこで、筆者らは、LED光という工学

的な見地から、ユーグレナにとって最適な光環境の追求を試みたものである。

すなわち、本論文は、ユーグレナに対して様々な光質条件のLED光を照射した際の酸素発生速度を測定し、さらに分析することで光合成速度を算出し研究結果をまとめたものである。具体的な光質条件としては、光量 (PPFD) の他に点滅周期や点滅デューティ比、光の波長に着眼した。まず初めに、光量を変化させた場合の光合成速度である「光-光合成曲線」を明らかにした。光源に赤色LEDを用いて、光量 (PPFD) を広範囲に変化させて「光-光合成曲線」を測定した。さらに、特定のPPFD値やデューティ比の条件下で、赤色LED点滅光と連続光での光合成速度を測定し比較した。

つぎに、点滅周期を変化させた場合の光合成速度特性「点滅周期-光合成曲線」を測定した。さらに、PPFD値による相互作用を検証するため、PPFD値を段階的に変更した場合の「点滅周期-光合成曲線」を測定し、PPFD値が「点滅周期-光合成曲線」へ及ぼす影響を調べた。つぎに、点滅デューティ比を変化させた場合の光合成速度特性である「点滅デューティ比-光合成曲線」を測定した。さらに、点滅周期による相互作用を検証するため、点滅周期を段階的に変更した場合の「点滅デューティ比-光合成曲線」を測定し、点滅周期が「点滅デューティ比-光合成曲線」へ及ぼす影響を調べた。

つぎに、2波長の光を混合して同時に照射することによる効果を検証するため、青色LED光を追加し、青色と赤色LEDの混合光を同時に照射した場合の「点滅周期-光合成曲線」と「点滅デューティ比-光合成曲線」を測定した。本章では、主要項目ごとにまとめ、若干の考察を加えたい。

5. 1. 光-光合成曲線について

ユーグレナを人工光で栽培する場合、「光-光合成曲線」は、光量 (PPFD) の設定値を決定する根拠となる重要な特性である。例えば、光量が光補償点よりも弱ければ光量不足で育たないし、光量が光飽和点よりも強ければ、光の利用効率悪化や光阻害の要因となるため、最適な光量を照射することが肝要になる。しかしながら、ユーグレナに関して「光-光合成曲線」の報告例はなかった。筆者らは、赤色LED連続光を用

いてユーグレナの「光-光合成曲線」を明らかにした。具体的には、光補償点や光飽和点や最適な光量の「光合成PPFD効率の最大点」を明らかにした。

5. 2. 点滅光と連続光との比較について

筆者らは、連続光の他に赤色LED点滅光を照射した場合の「光-光合成曲線」も明らかにした。植物に点滅光を照射した報告例は「サラダナ」に関する報告(森ら, 2002a, 2002b)があり、成長や光合成速度が促進されたと報告された。しかし、ユーグレナに点滅光を照射した報告はなかったため、筆者らは、ユーグレナに点滅光を照射し光合成速度への影響を明らかにした。具体的には、点滅光の点滅周期や点滅デューティ比や光量(PPFD)を変化させながら光合成速度を測定し「点滅周期-光合成曲線」や「点滅デューティ比-光合成曲線」を様々なPPFD値条件等で多角的に測定することで明らかにした。

その結果、点滅光での光合成速度と連続光での光合成速度を比較すると、両者は、おおよそ同等であった。従って、ユーグレナに関しては点滅光を照射する必要がなく連続光でも光合成速度が悪化しないと考えられる。また、点滅照射をした場合には、点滅周期をある周期(変化点)よりも長時間にすると光合成速度が急に低下することも明らかにした。従って、何らかの理由で点滅照射をする必要がある場合には、点滅周期をこの変化点以下に設定した方が光利用効率が良いと考えられる。

5. 3. 青色と赤色LED点滅光の同時照射について

筆者らは、青色と赤色LED点滅光を同時に照射した場合の「点滅周期-光合成曲線」や「点滅デューティ比-光合成曲線」も明らかにした。さらに、R/B比(赤と青の光量子数の比率)や点滅周期や点滅デューティ比に、成長や光合成速度の促進に効果があったと報告された値を採用し、これらの相乗効果等を狙い測定した。その結果、青色と赤色LED点滅光を同時に照射した場合と赤単色光を照射した場合を比較すると、両者は、おおよそ同じ光合成速度であった。但し、点滅周期をある周期よりも長時間に設定す

ると両者に違いが現れた。また、青赤混合照射による効果に関しては、青色光のPPFD比率を10%と低く設定したため両者の差が観察しにくかった可能性がある。

5. 4. まとめ

筆者らのこれまでの研究結果を表1にまとめた。表1からユーグレナ用照明としては基本的に、赤色の連続光で良いと考えられる。また、点滅光を照射する際には、点滅周期を変化点よりも短時間に設定する必要があると考えられる。

また、表1の項目「①」に記載した「光合成PPFD効率を最大にするPPFD値」は、大変大きな成果であり、太陽光を直接照射した場合と比較して光合成効率（光合成PPFD効率）を約22倍も向上させることが出来る特筆すべき成果である。過去に、このような「光合成PPFD効率」を報告した例はない。

従って、これらの研究成果は、極めて新規性が高い成果であり、ユーグレナにおける重要な学術的知見であると同時に、産業上の重要な設計技術情報を提供するものである。今後、「ユーグレナ用植物工場」等を建設する際に必要な高効率照明システムを新規に設計開発する上で非常に有益な技術情報である。

5. 5. 今後の課題

本研究の結果、酸素発生速度を指標とした光合成速度の短時間での評価は出来た。長期間の評価に関しては、既報（中野ら, 1998）にて「培養開始時に細胞濃度が急増し、数日で定常濃度に達した」と報告されたが、本研究の成果である「光合成PPFD効率を最大にするPPFD値」の長期間での評価は未検討であり、今後の課題として残されている。さらに将来、ユーグレナ用植物工場を開発するためには、光質条件制御の他に培地の成分制御や温度制御などの各種制御技術を発展させ応用する必要がある。そのためには、培養システムとして各技術を統合することが必要不可欠であると思われる。

表. 5 - 1 本論文で明らかにされた知見

	項 目	結 果
①	赤色 LED 連続光での「光-光合成曲線」	光合成 PPF 効率を最大にする PPF 値は、約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ で、光補償点は、約 $9 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ で、光飽和点は、約 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ だった (Fig. 2-15, 2-16)。
②	赤色 LED 点滅光と連続光との比較	両者の光合成速度は、おおよそ同等だった (Fig. 2-15, 2-16)。しかし、点滅周期を約 $5.0 \times 10^{-2} \text{ s}$ 以上にした場合、点滅光の光合成速度の方が低下することが明らかにされた (Fig. 2-17)。
③	赤色 LED にて各 PPF 値での「点滅周期-光合成曲線」	PPF 値 $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ で、点滅周期を約 $1.0 \times 10^{-2} \text{ s}$ 以上にすると急に光合成速度が低下した (Fig. 3-1)。PPF 値を高くすると、変化点（光合成速度が急に低下する周期）が短時間へ変化した (Fig. 3-1)。
④	赤色 LED での点滅デューティ比の影響	点滅デューティ比を変化させても光合成速度は、おおよそ変化しなかった (Fig. 3-2)。しかし、点滅周期を $1.0 \times 10^{-1} \text{ s}$ にした場合、デューティ比を下げるに伴い光合成速度が低下することが明らかにされた (Fig. 3-2)。
⑤	青色と赤色 LED の同時照射と赤単色照射の比較	同一 PPF 値での両者の光合成速度は、おおよそ同等だった (Fig. 4-5)。しかし、点滅周期を約 $3.0 \times 10^{-1} \text{ s}$ 以上にした場合、赤単色照射の光合成速度の方が低下することが明らかにされた (Fig. 4-5)。

基礎となる論文

- 1) 大橋光男、田中賢明、大井美知男：赤色LED点滅光が*Euglena*の光合成速度に及ぼす影響，生態工学会誌*Eco-Engineering*, 第23巻, 第4号, 123-129, 2011.

- 2) Mitsuo Ohashi, Souya Miyajima and Michio Ohi: Evaluation of the Effects of the Blinking Cycle and Duty Ratio of Red and Blue Light Emitting Diodes on the Photosynthetic Rate of *Euglena*, *Eco-Engineering*, 24(2), 43-49, 2012.

謝辞

本研究を行うにあたり、懇切丁寧な御指導と御鞭撻を賜りました主指導教員の信州大学農学部食料生産科学科 大井美知男 教授、及び副指導教員の中村宗一郎 教授、及び伴野潔 教授、及び的確な御助言を賜りました京都大学大学院農学研究科 土井元章 教授に心より御礼申し上げます。 また、貴重な *Euglena* 株を大変御親切に譲渡して頂き、さらに培養方法などの適切な御助言を多数賜りました、大阪府立大学名誉教授 宮武和孝 教授、及び大阪府立大学 中澤昌美 助教授に心より感謝申し上げます。 本研究は、KOA株式会社 代表取締役 向山孝一 社長、経営管理イニシアティブ 社会環境センター 田中賢明 ゼネラルマネージャー、ものづくりイニシアティブ 技作りセンター 宮島荘哉 ゼネラルマネージャー、赤羽秀樹 プロフィットマネージャー、及び社員の方々の寛大な御理解と御協力のもと行えたものであり、多大な御支援を頂きましたことを厚く御礼申し上げます。

最後に、私の研究に理解と協力を頂きました、妻子の大橋倍子、光希、由莉奈、直紀、沙央莉、及び両親の大橋米雄、芳子に感謝致します。 本当に、ありがとうございました。

引用文献

- Aiga, I., Murakami, K., Kiyota, M., Matsumoto, T., Yamaji, R., Nakano, Y., Miyatake, K., Takenaka, S., Masuda, W., Jinda, T., Horaguchi, K., Wasa, K. and Kondo, J., 1997: Growth of *Euglena*, a photosynthetic alga, by light emitting diodes. *CELSS J*, **9**(2), 7-12.
- Cramer, M. and Myers, J. C., 1952: Growth and photosynthetic characteristics of *Euglena gracilis*. *Arch. Microbiol.*, **17**, 384-402.
- 弟子丸幸子, 2012: 「ユーグレナ火力発電CO₂削減にミドリムシ」 『日経産業新聞』
2012/5/28, 日刊 (2)
- Donell, H. M., Jorgen, R. and Jorgen, R., 2005a: 成長の限界 人類の選択. ダイアモンド社, 東京. pp. 7.
- Donell, H. M., Jorgen, R. and Jorgen, R., 2005b: 成長の限界 人類の選択. ダイアモンド社, 東京. pp. 71-83.
- Donell, H. M., Jorgen, R. and Jorgen, R., 2005c: 成長の限界 人類の選択. ダイアモンド社, 東京. pp. 94-106.
- Emerson, R. and Rabinowitch, E., 1960 : Red drop and role of auxiliary pigments in photosynthesis. *Plant Physiol.*, **35**, 477-485.
- 藤田 智也・青柳 秀紀, 2012: 藻類を用いた有用物質の低負荷型生産. 用水と廃水, **54**(4), 304-312.
- 出雲 充, 2011: 21 世紀型の産学連携で初のミドリムシ培養に成功. 先見経済, **57**(4), 10-12.
- 出雲 充, 2012: 青年社長が「ミドリムシ」で世界を救う. *Themis*, **21**(3), 64-65.
- 北岡正三郎, 1989a: ユーグレナの細胞機能の解析と新規資源生物としての利用. 日本農芸化学会誌, **63**(11), 1741-1753.
- 北岡正三郎, 1989b: 序章. ユーグレナー生理と生化学ー (北岡正三郎編). 学会出版センター, 東京, pp. 1-21.

- Lincoln, T. and Eduardo, Z., 2004: 植物生理学 第3版. 株式会社培風館, 東京, pp. 109-141.
- 増田篤稔・洞口公俊・向坂信一・小澤知子・加藤元一・村上克介, 2006: 微細藻類の培養密度および培養液厚の変化に伴う透過光の減衰. *Eco-Engineering*, 18(1), 3-8.
- 松本隆仁・乾 博・宮武和孝・中野長久・村上克介, 2006: 光合成藻類ユーグレナの量子収率の検討. *Eco-Engineering*, 18(1), 21-25.
- 松本隆仁・乾 博・宮武和孝・中野長久・村上克介, 2007: 光合成培養におけるユーグレナの抗酸化ビタミン量への光質とCO₂濃度の影響. *Eco-Engineering*, 19(1), 33-38.
- Matsumoto, T., Inui, H., Miyatake, K., Nakano, Y. and Murakami, K., 2009: Comparison of nutrients in *Euglena* with those in other representative food sources. *Eco-Engineering*, 21(2), 81-86.
- 宮武和孝, 1999: 太陽光による廃炭酸ガスを利用した高度利用可能な効率的藻類バイオマス生産. 高温学会誌, 25(2), 66-70.
- 森康裕・高辻正基・安岡高志, 2002a: 白色LED パルス光がサラダナ育成に及ぼす影響. 植物工場学会誌, 14(3), 136-140.
- 森康裕・高辻正基・安岡高志, 2002b: 赤色LD パルス光が植物生育に及ぼす影響. レーザー研究, 30(10), 602-605.
- 長倉三郎・井口洋夫・江沢洋・岩村秀・佐藤文隆・久保亮五, 1998: 放射照度. 理化学事典 第5版. 岩波書店, 東京, pp. 1292.
- 中野良平・鈴木健吾・出雲充, 2012: 微細藻類ユーグレナを用いた環境技術の可能性. 用水と廃水, 54(4), 313-318.
- 中野長久・宮武和孝・乾博・穂波信雄・村上克介・金井謙二・辰巳雅彦・相賀一郎・近藤次郎, 1998: 地球環境を閉鎖・循環型生態系として配慮した食糧生産システム～藻類(ユーグレナ)の食糧資源化に関する研究～. CELSS 学会誌, 10(2), 13-23.

- 日本光合成研究会, 2003a: クロロフィルa/b 比. 光合成辞典. 株式会社学会出版センター, 東京, pp. 93.
- 日本光合成研究会, 2003b: 光合成速度. 光合成辞典. 株式会社学会出版センター, 東京, pp. 120.
- 日本光合成研究会, 2003c: 光合成の酸素阻害. 光合成辞典. 株式会社学会出版センター, 東京, pp. 124.
- 日本光合成研究会, 2003d: ヒル反応. 光合成辞典. 株式会社学会出版センター, 東京, pp. 290.
- 西澤潤一・垂井康夫・江刺正喜, 1999: 発光ダイオード. 半導体用語大辞典. 日刊工業新聞社, 東京, pp. 858.
- 佐伯佳浩・鈴木健吾・西浦英樹・田中克昌, 2008: 紫外線に対するユーグレナ酵素処理液の効果. *Fragrance Journal*, **36**(10), 76-80.
- 総務省, 2011a: 世界の人口の変化. 世界の統計., 総務省印刷局, 東京, pp. 28.
- 総務省, 2011b: 世界の人口の変化. 世界の統計., 総務省印刷局, 東京, pp. 107.
- Sugiyama, A., Suzuki, K., Mitra, S., Arashida, R., Yoshida, E., Nakano, R., Yabuta, Y. and Takeuchi, T., 2009: Hepatoprotective effects of paramylon, a β -1, 3-D-glucan isolated from *Euglena gracilis* Z, on acute liver injury induced by carbon tetrachloride in rats. *The Journal of Veterinary Medical Science*, **71**(7), 885-890.
- 高辻正基, 2006: 植物工場. 冷凍, **81**(948), 796-801.
- 高辻正基, 2007a: 植物工場とは. 完全制御型植物工場. 株式会社オーム社, 東京, pp. 1-21.
- 高辻正基, 2007b: 植物の光反応スペクトル. 完全制御型植物工場. 株式会社オーム社, 東京, pp. 27-28.
- 高辻正基・森 康裕, 2011: 照明設計上で重要なLEDの電気的特性. LED植物工場. 日刊工業新聞社, 東京, pp. 70-71.

和田正三・徳富 哲・長谷あきら・長谷部光泰, 2001: 気孔の青色光受容体. 植物の光センシング. 株式会社秀潤社, 東京, pp. 121-122.

山崎巖, 2011: 光合成の連鎖反応系をつくる各種タンパク質粒子の構成と機能. 光合成の光化学. 講談社, 東京, pp. 9-28.

柳 智博・岡本研正, 2004: 高輝度LEDを用いた植物栽培の可能性. 新技術コミュニケーションズ, 26(3), 295-300.

横田明徳, 1989: 葉緑体と光合成. ユーグレナー生理と生化学ー (北岡正三郎編). 学会出版センター, 東京, pp. 70-102.

出願特許

(1)大橋光男・相良岩男, 2006: LED駆動回路. 特願2006-283612, 特開2008-103470.

(2)大橋光男・小松英幸, 2007: LED駆動回路. 特願2007-24042, 特開2008-192730.

(3)大橋光男・相良岩男, 2007: LED駆動回路. 特願2007-40831, 特開2008-205276.

要旨

赤色及び青色LED点滅光の点滅周期とデューティ比が ユーグレナの光合成速度に及ぼす影響

本論文は、微細藻類であるユーグレナに様々な条件のLED点滅光を照射した場合の光合成速度について記述したものである。将来の人口爆発等により食糧不足が懸念される中、新たな食資源として、高い光合成効率や高栄養価等の優れた特長があるユーグレナ (*Euglena*) という藻類の活用が提案されている。本論文では、ユーグレナの効率的な栽培を目指し、様々な条件のLED点滅光をユーグレナに照射した場合の光合成速度を測定することで、「光-光合成曲線」などの重要なデータを明らかにすることを目的とした。

第1章では、研究の背景、及び植物工場やLED点滅光照射下での栽培に関する既往の知見などを確認した。

第2章では、ユーグレナに赤色LED点滅光を照射した場合の「光-光合成曲線」や「点滅光周期の光合成速度に対する影響」などの研究を行い、ユーグレナに適した点滅周期や光量を明らかにした。この結果から赤色LEDを用いてユーグレナ用の効率的な光源装置を考える場合、PPFD値を「約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 」とし、さらにLEDの点滅周期を「約 $5.0 \times 10^{-2}\text{s}$ よりも短時間」とすれば、光合成効率が低下しないことが分かった。補足だが、点滅点灯での光合成速度を正確に測定するためには、「10 ns 単位」で精密に点滅タイミングを制御可能であり、さらに点灯時の光量を正確に調整可能な光源ユニットが必要である。しかし、この条件を満足する市販品の光源ユニットは、なかった。従って、新規に光源ユニットを電子回路から設計し製作した。

第3章では、点滅光の「デューティ比」や「PPFD値」及び「点滅光の周期」を変化させ、光合成速度に対する影響を測定した。この結果から赤色LEDを用いてユーグレナ用の効率的な光源装置を考える場合、LED照射光のPPFD値が $400 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 以下

の条件であれば、「点滅周期は 1.0×10^{-2} s よりも短時間にすればよい」ことと、PPFD 値を光合成PPFD効率が低い「約 $35 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 」の条件とした場合、「デューティ比の設定範囲を10 ~ 100 % にすればよい」ことが分かった。

第4章では、異なる波長の光（青色光と赤色光）を同時に照射した場合の光合成速度に対する影響を測定した。この結果から「最適周期領域」にて、点滅デューティ比を33 % とした場合には、「LEDの点滅周期を約 5.0×10^{-3} s 以下」にすると光合成効率がよく維持されることが分かった。さらに、青色LEDと赤色LED光の混合照射条件下にて点滅周期を 5.0×10^{-1} s とした場合には、点滅デューティ比を10 ~ 100 % の範囲で変化させても光合成効率が低下しないことが分かった。

第5章では、総括として上述の研究成果をまとめた。特に、第2章で得られた成果である「光合成PPFD効率を最大にする光量条件（Point-C）」での「光合成PPFD効率」（ η_{max} ）を算出した。更に、この値と太陽光と同じ光量で照射した場合の「光合成PPFD効率」（ η_{sun} ）を比較した。その結果、「 η_{max} 」の方が「 η_{sun} 」よりも約22倍も光合成PPFD効率が高くなることが分かった。すなわち、太陽光を直接照射して培養するよりも、光量を Point-C の PPFD値に制御して照射することで、約22倍も光合成PPFD効率を向上させることが出来るということである。

この成果は、人工光での培養に応用可能であるばかりでなく、太陽光を用いる場合にも応用可能である。例えば、太陽光を「シリンドリカルレンズと導光板で培養槽へ照射する方式」（宮武, 1999）を応用して、Point-C の光量値に分割し培養槽へ照射することで、光合成PPFD効率を約22倍も向上させることが出来るということである。

これらの研究成果は、極めて新規性が高い成果であり、ユーグレナの光合成に関する重要な基礎学術的知見を提供すると同時に、今後「ユーグレナ用植物工場」などを開発する上で非常に有益な技術情報である。

付 録

付録目次

付録1：プログラムリスト

付録2：LEDドライブ部の回路動作

付録1. プログラムリスト

List. 1 - 1 Main program

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```

/*****
LED 電流制御基板 U4(PIC18F452)
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
*****/

#define PIC18
#include <pic18.h>
    _CONFIG(1, OSCSDIS & ECRA6); //クロックセレクト無効、OSC2はRA6として使用。
    _CONFIG(2, WDTDIS & PWRTEEN & BOREN & BORV45); //WDT=無効、パワータイマー有効、BOR有効(4.5V)
    _CONFIG(3, CCP2RC1); //CCP2 i/o multiplexed with RC1
    _CONFIG(4, DEBUGDIS & LVPDIS & STVREN); //デバッグ無効、低電圧書き込み無効、スタックリセット有効

#include <string.h>
#include "Delay.h"
#include "Eeprom_led.h" //LED基板用
#include "Rs232c.h"
#include "Command.h"
#include "Time.h"
#include "LED_PNL.h"

void DeviceInit(void);
void Display(void);
void DA_WRT(char, char);
void DA_INIT(void);
void I2CInit();
char IO_WRT(char, char);
char IO_RED(char);
void IO_INIT(void);
void OUT_7SEG_LED(unsigned char);
char READ_DIPSW(void);
void OUT_LEDS(char); //表示用LED(D2~5)にCH値を表示

char TEST_EEPROM(void);
char WRITE_U8(unsigned char); //U8にデータを書き込む
char WRITE_U9(unsigned char); //U9にデータを書き込む
char READ_EEP_SET_AD(void); //EEPROMを読み、A/Dなどに書き込む。
char SET_DEFAULT_EEPROM(void); //EEPROMにデフォルト値を代入する。
char SET_WRITING_EEPROM(void);
char SET_VALID_EEPROM(void);

```



```

char CHK_EEPROM(void);
char WRITE_EEPROM(unsigned int ,unsigned char);
unsigned char READ_EEPROM(unsigned int );
void DEB_1(void);
char CHK_DIPSW_SET_ARY(void); //DIP-SWの操作で、電流設定用変数を変更する。
char CHK_DIPSW_SET_VAL_IO(void); //DIP-SWの操作で、I/O設定用変数を変更する。
char SET_DATA0_TO_1(void); //変数DATA1やIO_U8_DATA1にDATA0等の値を代入する。
char SET_ARY_TO_EEPROM(void); //変数xx0の値をEEPROMに書き込む
(M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
char CHK_EEPROM_SET_ARY0(void); //EEPROMに有効な数値が有れば変数0に代入。
void OUT_7SEG_LED_ISEC(unsigned char); //OUT TO 7SEG LED WITH WAIT 1 sec.

unsigned char CHO_200mA, CHO_600mA;
unsigned char CH1_200mA, CH1_600mA;
unsigned char CH2_200mA, CH2_600mA;
unsigned char CH3_200mA, CH3_600mA;
unsigned char CH4_200mA, CH4_600mA;
unsigned char CH5_200mA, CH5_600mA;

unsigned char M_NAGAOSI; //長押し処理用変数
unsigned char M_DELTA; //変数の変化分の絶対値
unsigned char M_AD_DATA0[8]; //A/D値の作業変数（これから書き込むデータ用）
unsigned char M_AD_DATA1[8]; //A/D値の直前の設定値保存用
unsigned char M_CH; //CH値の初期値設定（0～5=各ch、6=予備、7=全CH指定）
unsigned char IO_U8_DATA0; //I2C_IO U8の作業変数（これから書き込むデータ用）
unsigned char IO_U8_DATA1; //I2C_IO U8の直前の設定値保存用
unsigned char IO_U9_DATA0; //I2C_IO U9の作業変数（これから書き込むデータ用）
unsigned char IO_U9_DATA1; //I2C_IO U9の直前の設定値保存用
unsigned char M_DIPSW0; //DIP SW の読込み値（現在）
unsigned char M_DIPSW1; //DIP SW の読込み値（前回）

/*=====
main loop
=====*/
main(void)
{
    unsigned char i, j, k, l, m, n, adr, data;

    //Initialise CPU
    DeviceInit(); //I/Oポート初期化
    XOE = 1; //全LED消灯
    DA_INIT(); //D/Aコンバータ初期化（出力電圧=0）
    IO_INIT(); //I/Oエキスパンダー初期化

    //-----EEPROMを確認し有効な数値が有れば変数代入しなければデフォルト値を代入する。
    CHK_EEPROM_SET_ARY0(); //EEPROMに有効な数値が有れば変数に代入。
    //-----変数にダミー値を代入する-----
    // 最初にDA等に書き込むため。
    for(j=0; j<8; j++){
        M_AD_DATA1[j] = ~M_AD_DATA0[j]; //反転の値を代入
    }
    IO_U8_DATA1 = IO_U8_DATA0;
    IO_U9_DATA1 = IO_U9_DATA0;
    M_CH = 7; //CH値の初期値設定（0～5=各ch、6=予備、7=全CH指定）

```

```

M_DIPSW0 = READ_DIPSW0(); //今のDIP-SW値を読み込む。
//-----
DelaySec(1); //ループに入る前にダミー待ち。
CHK_DIPSW_SET_ARY(); //DIP-SWの操作で、変数M_AD_DATA0[]等を変更する。
OUT_LEDS(M_CH); //表示用LED (D2～5)にCH値を表示
OUT_7SEG_LED(M_AD_DATA0[M_CH]); //7SEG LEDに表示
CHK_DIPSW_SET_VAL_IO(); //M_DIPSW0の値で、I/O用変数 (IO_U8_DATA0とIO_U8_DATA0)を変更する。
SET_ARY_TO_EEPROM(); //変数??0の値をEEPROMに書き込む (M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
READ_EEP_SET_AD0(); //EEPROMを読み、A/Dなどに書き込む。
SET_DATA0_TO_I0(); //変数DATA1やIO_U8_DATA1にDATA0等の値を代入する。

//====MAIN LOOP=====

for( ; ; ){ //メインLOOP (周期約1秒)

//-----DIP-SWによる変更を変数0に代入する (M_AD_DATA0[]など)-----
CHK_DIPSW_SET_ARY(); //DIP-SWの操作で、変数M_AD_DATA0[]等を変更する。
OUT_LEDS(M_CH); //表示用LED (D2～5)にCH値を表示
OUT_7SEG_LED(M_AD_DATA0[M_CH]); //7SEG LEDに表示
CHK_DIPSW_SET_VAL_IO(); //M_DIPSW0の値で、I/O用変数 (IO_U8_DATA0とIO_U8_DATA0)を変更する。
//-----通信による変更を変数に代入する (M_AD_DATA0[]など)-----
//-----変数0と変数1 (EEPROMと同値)を比較する。-----
i=0; //判定用フラグを初期化。(同じなら0)
for(j=0; j<6; j++){
if (M_AD_DATA1[j] != M_AD_DATA0[j]){
i=1; //判定フラグを「異なった」とする。
}
}
if (IO_U8_DATA1 != IO_U8_DATA0) {i=1;} //判定フラグを「異なった」とする。
if (IO_U9_DATA1 != IO_U9_DATA0) {i=1;} //判定フラグを「異なった」とする。

//-----変更があった場合に以下の段落を処理-----
if (0 != i){
//-----全LED消灯-----
XOE = 1; //全LED消灯
//-----変数0の値をEEPROMに書き込む。-----
SET_ARY_TO_EEPROM(); //変数0の値をEEPROMに書き込む
(M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
//-----EEPROMの値をD/AやI/Oに書き込む。-----
READ_EEP_SET_AD0(); //EEPROMを読み、A/Dなどに書き込む。
//-----変数1に変数0を代入する。-----
SET_DATA0_TO_I0(); //変数DATA1やIO_U8_DATA1にDATA0等の値を代入する。
//-----全LED点灯-----
XOE = 0; //全LED点灯
}
//-----LOOP時間調整-----
DelaySec(1); //ループ時間を1秒に設定。
//-----メインLOOPの先頭に戻る-----
} //MAIN ループの終点
} //MAIN関数の終り

/*-----
DIP-SWの操作で、I/O設定用変数を変更する。
IN:M_DIPSW0

```

```

OUT:IO_U8_DATA0, IO_U9_DATA0.
=====*/
char CHK_DIPSW_SET_VAL_10(void) //DIP-SWの操作で、I/O設定用変数を変更する。
{
    if (0x08 & M_DIPSW0) {
        CH0_200mA = 1;
        CH1_200mA = 1;
        CH2_200mA = 1;
        CH3_200mA = 1;
        CH4_200mA = 1;
        CH5_200mA = 1;
    }
    else {
        CH0_200mA = 0;
        CH1_200mA = 0;
        CH2_200mA = 0;
        CH3_200mA = 0;
        CH4_200mA = 0;
        CH5_200mA = 0;
    }
    if (0x10 & M_DIPSW0) {
        CH0_600mA = 1;
        CH1_600mA = 1;
        CH2_600mA = 1;
        CH3_600mA = 1;
        CH4_600mA = 1;
        CH5_600mA = 1;
    }
    else {
        CH0_600mA = 0;
        CH1_600mA = 0;
        CH2_600mA = 0;
        CH3_600mA = 0;
        CH4_600mA = 0;
        CH5_600mA = 0;
    }
    //--- U 8用のデータ作成-----
    IO_U8_DATA0 = CH3_200mA + (CH2_200mA << 1) //各ビットを組み合わせて、8bitにする。
                + (CH1_200mA << 2) + (CH0_200mA << 3)
                + (CH0_600mA << 4) + (CH1_600mA << 5)
                + (CH2_600mA << 6) + (CH3_600mA << 7);
    //--- U 9用のデータ作成-----
    IO_U9_DATA0 = CH5_200mA + (CH5_600mA << 1) //各ビットを組み合わせて、8bitにする。
                + (CH4_200mA << 2) + (CH4_600mA << 3)
                + 0xF0; //D_SW0~3は入力なので、1を書く。
    //-----ここまでで、IO_U8_DATA0とIO_U9_DATA0は、代入完了-----
}
/*-----*/
DIP-SWの操作で、電流設定用変数を変更する。
IN:なし
OUT:M_DIPSW0, M_DIPSW1, M_NAGAOS1, M_AD_DATA0[], M_AD_DATA1[],
=====*/
char CHK_DIPSW_SET_ARY(void) //DIP-SWの操作で、電流設定用変数を変更する。
{

```

```

// <DIP-SWによる動作定義> (但し、SWのデフォルトはON(値=0)とする。)
// BIT0=UP (数値を増加する) (通常は1毎に変化し、10秒間以上長押しすると、10H毎になる。)
// BIT1=DOWN (数値を減少する) (同上)
// BIT2=変数選択用 (ON=A/D値、OFF=CH指定値)
// BIT3=電流レンジ設定 (中)
// BIT4=電流レンジ設定 (大)
//まず、BIT2を読み、A/DかCHかの処理を分ける。

unsigned char i, j;

M_DIPSW1 = M_DIPSW0; //前回の読み込み値を保存
M_DIPSW0 = READ_DIPSW(); //今のDIP-SW値を読み込む。
if (0x04 & M_DIPSW0) { //真ならCH指定
    if (0x01 & M_DIPSW0) {M_CH += 1;} //UP-SW押された場合。
    if (0x02 & M_DIPSW0) {M_CH -= 1;} //DOWN-SW押された場合。
    if (0x07 < M_CH) {M_CH = 0;} //c h値の最大を7に制限
    M_NAGAOSI=0; //長押し用変数クリア
}

else //偽ならA/D値指定
{
    if ( (0x03 & M_DIPSW0) && !(0x04 & M_DIPSW0) && (M_DIPSW1 == M_DIPSW0) ){
        M_NAGAOSI++; //A/D値指定で、かつ (UP又はDOWN) かつ (SWの現在値=前回値) ならば++。
    } else {M_NAGAOSI = 0;} //以外なら、長押し変数クリア。

    if (7 < M_NAGAOSI) {M_DELTA=0x10;} //長押しなら、10Hづつ増減。
    else {M_DELTA=1;} //非長押しなら、1づつ増減。

    if (0x01 & M_DIPSW0) { //UP-SW押された場合。
        if (7 == M_CH) { //全CH指定の場合
            M_AD_DATA0[M_CH] += M_DELTA; //変数を増加する。
            for(j=0; j<6; j++){ //各CH変数に代入する。
                M_AD_DATA0[j] = M_AD_DATA0[M_CH];
            }
        }
        else { //非全CHの場合 (個別のCHを指定)
            M_AD_DATA0[M_CH] += M_DELTA; //変数を増加する。
        }
    }

    if (0x02 & M_DIPSW0) { //DOWN-SW押された場合。
        if (7 == M_CH) { //全CH指定の場合
            M_AD_DATA0[M_CH] -= M_DELTA; //変数を一する。
            for(j=0; j<6; j++){ //各CH変数に代入する。
                M_AD_DATA0[j] = M_AD_DATA0[M_CH];
            }
        }
        else { //非全CHの場合 (個別のCHを指定)
            M_AD_DATA0[M_CH] -= M_DELTA; //変数を増加する。
        }
    }
}
}

/*=====

```

電源投入時に、EEPROMを確認し有効な数値が有れば変数0に代入し、なければデフォルト値を代入する。

```

IN:なし
OUT:戻り値 (0)
=====*/
char CHK_EEPROM_SET_ARY0(void) //EEPROMに有効な数値が有れば変数0に代入
{
    unsigned char i=0, j=0;

    if (0 != CHK_EEPROM()) { //EEPROMの内容が有効か否かを検査 (0なら有効)
        //もしも無効なら、デフォルト値をEEPROMに書く。
        SET_DEFAULT_EEPROM(); //EEPROMにデフォルト値を代入する。
        i=1; //無効な値の処理をしたので、フラグを立てる。
    }

    //EEPROMから読み、変数0に代入する。
    IO_U8_DATA0 = READ_EEPROM(TBL_U8_EEPROM);
    IO_U9_DATA0 = READ_EEPROM(TBL_U9_EEPROM);
    for(j=0; j<6; j++){
        M_AD_DATA0[j] = READ_EEPROM(TBL_TOP_EEPROM+j);
    }
    M_AD_DATA0[7] = M_AD_DATA0[0]; //EEPROM領域に[7]のデータはないので、仮に代入する。(単色対応)
    return i;
}

```

```

/*=====
EEPROMの書き込み&読み込み検査を行う。
もしも、NGなら消灯しSTOPする。(基板上のLEDを点滅させる。)
用途は、EEPROM検査関数(CHK_EEPROM())にて使用する。
IN:なし
OUT:戻り値 (0:OK、0以外:エラー有り)
=====*/

```

```

char TEST_EEPROM(void)
{
    unsigned char err0 = 0;
    unsigned char i;
    //検査パターンその1
    for (i=0; i != 0xFF; i++){ WRITE_EEPROM(i, i); } //まず、0から順にデータを書く
    for (i=0; i != 0xFF; i++){
        if (i != READ_EEPROM(i)){ err0 =1; } //もしも、書き込みデータ以外を読んだらエラーとする。
        OUT_LEDS(i); //表示用LED (D2~5)に表示
    }
    //検査パターンその2
    for (i=0; i != 0xFF; i++){ WRITE_EEPROM(i, (~i)); } //反転パターンを順に書く
    for (i=0; i != 0xFF; i++){
        if (i != (0xFF & ~READ_EEPROM(i))){ err0 =2; } //もしも、書き込みデータ以外を読んだらエラーとする。
        OUT_LEDS(~i); //表示用LED (D2~5)に表示
    }
    //エラー処理
    for ( ; 0 != err0; ){ //異常を検出したら、基板上のLEDを点滅しながら、無限LOOPする。
        XOE = 1; //全LED消灯
        OUT_LEDS(0x00); //表示用LED (D2~5)に表示
        DelayMs(250);
        OUT_LEDS(0xFF); //表示用LED (D2~5)に表示
    }
}

```

```

        DelayMs(250);
    }
    return err0;
}

/*=====
U 8にデータを書き込む
IN:書き込みデータ (1バイト)
OUT:戻り値 (0)
=====*/
char WRITE_U8(unsigned char i ) //U 8にデータを書き込む
{
    i = i & 0x0F; //上位4ビット (600mA) をマスクする。(安全のため)
    IO_WRT(I2C_I0_U8, i); //EEPROMからデータを読み、A/Dに書き込む。
    return 0;
}

/*=====
U 9にデータを書き込む
IN:書き込みデータ (1バイト)
OUT:戻り値 (0)
=====*/
char WRITE_U9(unsigned char i ) //U 9にデータを書き込む
{
    i = i & 0xF5; //ビット1と3 (600mA) をマスクする。(安全のため)
    i = i | 0xF0; //上位4ビットを1にする。(入力のため)
    IO_WRT(I2C_I0_U9, i); //EEPROMからデータを読み、A/Dに書き込む。
    return 0;
}

/*=====
EEPROMを読み、A/Dなどに書き込む。
IN:なし
OUT:戻り値 (0)
ADR 00,01,02,03,04,05,06,07,08,09, 0A, 0B, 0C, 0D, 0E, 0F
DATA F, W, W, W, -, -, -, -, U8,U9,AD0,AD1,AD2,AD3,AD4,AD5
=====*/
char READ_EEP_SET_AD(void) //EEPROMを読み、A/Dなどに書き込む。
{
    unsigned char j;

    XOE = 1; //全LED消灯
    for(j=0; j<6 ; j++){
        DA_WRT(j, READ_EEPROM(TBL_TOP_EEPROM+j));} //EEPROMから6ch分を読みD/Aへ書き込む。(ch, data)
    WRITE_U8(READ_EEPROM(TBL_U8_EEPROM)); //EEPROMからデータを読み、A/Dに書き込む。
    WRITE_U9(READ_EEPROM(TBL_U9_EEPROM)); //EEPROMからデータを読み、A/Dに書き込む。
    XOE = 0; //全LED点灯
    return 0;
}

/*=====
変数0の値を変数1に代入する。
IN:なし
OUT:戻り値 (0)
=====*/
char SET_DATA0_TO_1(void) //変数DATA1やIO_U8_DATA1にDATA0等の値を代入する。
{

```

```

    unsigned char j;
    for(j=0; j<8; j++){
        M_AD_DATA1[j] = M_AD_DATA0[j];
    }

    IO_U8_DATA1 = IO_U8_DATA0;
    IO_U9_DATA1 = IO_U9_DATA0;
    return 0;
}

/*=====
EEPROMにデフォルト値を代入する。
IN:なし
OUT:戻り値 (0)
=====*/
char SET_DEFAULT_EEPROM(void) //EEPROMにデフォルト値を代入する。
{
    unsigned char j;
    for(j=0; j<8; j++){ //A/D設定変数を初期化 (MAX値)
        M_AD_DATA0[j] = 0x80; //初期値をMAX値とする。
    }

    IO_U8_DATA0 = 0x00;
    IO_U9_DATA0 = 0xF0;
    SET_DATA0_TO_10; //変数DATA1やIO_U8_DATA1にDATA0等の値を代入する。
    SET_ARY_TO_EEPROM(); //EEPROMに書き込む (M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
    return 0;
}

/*=====
変数0の値をEEPROMに書き込む (M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
IN:なし
OUT:戻り値 (0)
ADR 00,01,02,03,04,05,06,07,08,09, 0A, 0B, 0C, 0D, 0E, 0F
DATA F, W, W, W, -, -, -, U8, U9, AD0, AD1, AD2, AD3, AD4, AD5
=====*/
char SET_ARY_TO_EEPROM(void) //変数0の値をEEPROMに書き込む
(M_AD_DATA0[], IO_U8_DATA0, IO_U9_DATA0)
{
    unsigned char j;
    SET_WRITING_EEPROM(); //フラグを「書き込み中」にする。
    WRITE_EEPROM(TBL_U8_EEPROM, IO_U8_DATA0); //U8用
    WRITE_EEPROM(TBL_U9_EEPROM, IO_U9_DATA0); //U9用
    for(j=0; j<6; j++){ //A/D設定変数を初期化 (MAX値)
        WRITE_EEPROM(TBL_TOP_EEPROM+j, M_AD_DATA0[j]); //EEPROMへ書き込む、初期値を80とする。
    }
    SET_VALID_EEPROM(); //EEPROMの内容が、有効である、という印を書き込む。(A5, AA, 55)
    return 0;
}

/*=====
EEPROMへ書き込み中のフラグを立てる。
IN:なし
OUT:戻り値 (0)
=====*/
char SET_WRITING_EEPROM(void)
{
    WRITE_EEPROM(FLG_EEPROM, 0x01); //フラグを「書き込み中」にする。
    return 0;
}

```

```

/*=====
EEPROMの内容が、有効である、という印を書き込む。(有効の印=A5, AA, 55)
IN:なし
OUT:戻り値 (0)
=====*/
char SET_VALID_EEPROM(void)
{
    SET_WRITING_EEPROM(); //フラグを「書き込み中」にする。
    WRITE_EEPROM(CHK_EEPROM0, 0x55); //有効の印を書く (A5, AA, 55)
    WRITE_EEPROM(CHK_EEPROM1, 0xAA);
    WRITE_EEPROM(CHK_EEPROM2, 0xA5);
    WRITE_EEPROM(FLG_EEPROM, 0x00); //フラグを「書き込み完了」にする。
    return 0;
}

/*=====
リセット時に、EEPROMの内容が、有効なデータか否かを判断する。
IN:なし
OUT:戻り値 (0:有効、0以外:無効なデータ)
=====*/
char CHK_EEPROM(void)
{
    unsigned char i =0;
    if (0x00 != READ_EEPROM(FLG_EEPROM)) {i=1;} //EEPROM用フラグ (0:通常、1:書き込み中)
    if (0x55 != READ_EEPROM(CHK_EEPROM0)) {i=1;} //有効性判定用 (A5, AA, 55)
    if (0xAA != READ_EEPROM(CHK_EEPROM1)) {i=1;}
    if (0xA5 != READ_EEPROM(CHK_EEPROM2)) {i=1;}
    // もしも、無効なデータなら、EEPROMのW/R検査を実施する。
    if (0 != i) {TEST_EEPROM();} //EEPROMの書き込み&読み込み検査を行う。(NGなら、STOPする。
    return i;
}

/*=====
LED基板のEEPROMへデータを書き込む
IN:WADRメモリー上のアドレス (0~0xFFFF)、書き込むデータ(0~0xFF)
OUT:戻り値 (読んだデータ(BYTE))
=====*/
char WRITE_EEPROM(unsigned int WADR1, unsigned char DATA1)
{
    if (DATA1 != READ_EEPROM(WADR1)) { //現在値と書き込み値が異なる場合のみ、書き込む。
        I2CDeviceCode = LED_ROM_P0; //デバイスコードを設定
        for( ; 0 != I2CPutByte(WADR1, DATA1); ) {} //成功するまでloopする。
    }
    return 0;
}

/*=====
LED基板のEEPROMからデータを読み込む
IN:WADRメモリー上のアドレス (0~0xFFFF)
OUT:戻り値 (読んだデータ(BYTE))
=====*/
unsigned char READ_EEPROM(unsigned int WADR1)
{
    I2CDeviceCode = LED_ROM_P0;
    return I2CGetByte(WADR1);
}

//=====

```



```

void DEB_1(void)
{
    unsigned char i;
    //まず、0番地から順に書く
    for (i=0; i<0xff; i++){
        WRITE_EEPROM(i, i);
    }
    //つぎに、0番地から順に読み、表示する。
    for (i=0; i<0xff; i++){
        OUT_7SEG_LED(0xAA);           //OUT TO 7SEG LED
        DelaySec(1);                  //1秒
        OUT_7SEG_LED(READ_EEPROM(i)); //OUT TO 7SEG LED
        DelaySec(1);                  //1秒
    }
    for ( ; ; ){} //無限LOOP
}

/*=====
READ DIP SW
=====*/
char READ_DIPSW(void)
{
    char i;
    i = (IO_RED(I2C_IO_U9)>>4) & 0xF; //RD5 << 4);
    i = ~i & 0x1F; //論理を反転する。
    return i;
}

/*=====
TEST IO_WRT(IC, DATA)
=====*/
void TEST_IO_WRT(void)
{
    char j, i;
    for(j=1; j<=10; j++){
        for(i=0; i < 0xFF; i++){
            DelayMs(1);
            RD7=0;
            DelayMs(1);
            RD7=1;
            IO_WRT(I2C_IO_U8, i); //IN:adr,Write data OUT:Status(!=0 then error)
        }
        DelaySec(1);
        RD7=0;
        DelaySec(1);
        RD7=1;
        for(i=0; i < 0xFF; i++){
            DelayMs(1);
            RD7=0;
            DelayMs(1);
            RD7=1;
            IO_WRT(I2C_IO_U9, i); //IN:adr,Write data OUT:Status(!=0 then error)
        }
    }
}

/*=====

```

付録-13

```

OUT_TSEGMENT_LED WITH WAIT 1 sec
=====
void OUT_TSEG_LED_ISEC(unsigned char data) //OUT TO TSEG_LED WITH WAIT 1 sec.
{
    OUT_TSEG_LED(data); //OUT TO TSEG_LED
    DelaySec(1);
}
=====
OUT_TSEGMENT_LED
=====
void OUT_TSEG_LED(unsigned char data) //OUT TO TSEG_LED
{
    PORTB = data; //OUT PORTB
}
=====
Init I/O EXPANDER PCF8574
=====
void IO_INIT(void)
{
    I2CInit(); // I 2 C (CPU側) 初期化
    I0_WRT(12C_10_18, 0x00); //IN:addr, Write data OUT:Status(!=0 then error)
    I0_WRT(12C_10_19, 0x00); //IN:addr, Write data OUT:Status(!=0 then error)
}
=====
TEST of DA_WRT(AD_CH, AD_DATA)
=====
void TEST_DA_WRT(void)
{
    char i, j;
    for(j=0; j<8; j++) {
        RD7=0;
        DelaySec(1);
        RD7=1;
        DelaySec(1);
        for(i=0; i<255; i++) {
            DA_WRT(j, i); //ch, data
        }
    }
}
=====
Write D/A CONVERTER MB8347 (D/A converter 8bit) OK
=====
void DA_WRT(AD_CH, AD_DATA) //0-7
{
    char AD_CH; //0-7
    char AD_DATA; //0-0xFF
}
char IC_CH, ZCH, ZDATA, i;
if (AD_CH > 7) [AD_CH = 7:] //CH設定値が範囲外だと、7にする。
switch (AD_CH) { //1 Cのピンアサインの変換
case 0:
    IC_CH = 5;
    break;
case 1:
    IC_CH = 6;
}
}
=====

```

```

        break;
    case 2:
        IC_CH = 3;
        break;
    case 3:
        IC_CH = 4;
        break;
    case 4:
        IC_CH = 1;
        break;
    case 5:
        IC_CH = 2;
        break;
    case 6:
        IC_CH = 7;
        break;
    case 7:
        IC_CH = 8;
        break;
    default:
        IC_CH = 8;
        break;
}
do { //設定値と読み込み値が一致するまで、リトライする。
    DelayUs(1);
    ZCH = IC_CH;
    //アドレス部送信 (4ビット)
    for (i=0; i<4; i++){
        DA_DI = ZCH & 0x01; //ビット代入、大丈夫?
        ZCH >>= 1;
        DA_CLK = 1;
        DelayUs(1);
        DA_CLK = 0;
        DelayUs(1);
    }
    //データ部送信 (8ビット)
    ZDATA = AD_DATA;
    for (i=0; i<8; i++){
        DA_DI = ((ZDATA & 0x80) >> 7);
        ZDATA <<= 1;
        DA_CLK = 1;
        DelayUs(1);
        DA_CLK = 0;
        DelayUs(1);
    }
    //A/Dレジスタへロード (LD信号↑↓)
    DA_LD = 1;
    DelayUs(1);
    DA_LD = 0;
    DelayUs(1);
    //アドレス読み込み
    ZCH = 0; //読み込み変数クリア
    for (i=0; i<4; i++){
        ZCH >>= 1;

```

```

// DI=CLK=LD=0
DA_DI = 0;
DA_CLK = 0;
DA_LD = 0;

char j;
*/
11) If not same then retry.
10) If same then OK.
9) Check send data and store data.
8) Loop 12 times from 6).
7) Set CLK=1 next 0
6) Load DO and store
5) Set LD=1 next 0
4) Loop 12 times from 2).
3) Set CLK=1 next 0
2) Set DI
1) Make send data of 12bit
Data set flow:
Data format: (start)D0-D7,A3-A0(end)
LD: IN: load data to D/A reg. (positiv edg) (set low normal)
CLK: IN: CLK input (positiv edg) (set low normal)
DO: OUT: output serial data
DI: IN: input serial data (12bit) (set low normal)
*/ MAX_frg of CLK = 2.5MHz
{
void DA_INIT(void)
=====
/* Init D/A CONVERTER MB88347(D/A converter 8bit)
=====
} //main
} while ( !(ZCH == IC_CH) && (ZDATA == AD_DATA) );
}
Delays(1);
DA_CLK = 0;
Delays(1);
DA_CLK = 1;
[ZDATA = ZDATA & 0xFE;]
else
[ZDATA = ZDATA | 0x01;]
if (1 == DA_LD)
ZDATA <<= 1;
for (i=0; i<8; i++) {
ZDATA = 0;
// 読み込み変数クリア
// 1ビット読み込み
ZCH = ZCH & 0x0F;
// 上位ビットをマスクする。
Delays(1);
DA_CLK = 0;
Delays(1);
DA_CLK = 1;
[ZCH = ZCH & 0xF7;]
else
[ZCH = ZCH | 0x08;]
if (1 == DA_LD)
}

```

```

//-----全CHを0に初期化する。-----
for(j=0; j<8 ; j++){
    DA_WRT(j, 0);          //(ch, data)
}
}

/*=====
Init Device
=====*/
void DeviceInit(void)
{
    SCS = 0;                //System Clock Switch (SELECT CLK PIN)
// --- RCON ---
    IPEN = 1;              //Interrupt Priority Enable(RCON-BIT7)
// --- INTCON ---
    GIEH = 0;              //HIGH Interrupt DIS Enable
    GIEL = 0;              //LOW Interrupt DIS Enable
// --- INTCON2 ---
    RBPU = 1;              //Port B Pull-Up Enable
// --- PIE1 ---
    PSPIE = 0;             //Parallel Slave Port Read/Write Interrupt
    ADIE = 0;              //AD Converter Interrupt
    RCIE = 0;              //USART Recieve Interrupt
    TXIE = 0;              //USART Transmit Interrupt
    SSPIE = 0;             //Master Synchronous Serial Port Interrupt
    CCP1IE = 0;            //CCP1 Interrupt
    TMR2IE = 0;            //TMR2 Overflow Interrupt
    TMR1IE = 0;            //TMR1 Overflow Interrupt
// --- PIE2 ---
    EEIE = 0;              //EEPROM/FLASH Write Interrupt
    BCLIE = 0;             //Bus Collision Interrupt
    LVDIE = 0;             //Low Voltage Detect
    TMR3IE = 0;            //TMR3 Overflow Interrupt
    CCP2IE = 0;            //CCP2 Interrupt
// --- PORTA ---(SET TRISTATE)
    PORTA = 0XFF;          //CLR PORT A
    TRISA = 0XFF;          //TRISTATE
// --- PORTB ---(SET OUT)
    PORTB = 0XFF;          //CLR PORT
    TRISB = 0X00;          //OUT
// --- PORTC ---(SET INPUT)
    PORTC = 0XFF;          //CLR PORT
//    TRISC = 0XFF;          //TRISTATE
    TRISC7 = 1;            //TRISTATE
    TRISC6 = 1;            //TRISTATE
    TRISC5 = 0;            //OUT
    TRISC4 = 1;            //TRISTATE
    TRISC3 = 1;            //TRISTATE
    TRISC2 = 1;            //TRISTATE
    TRISC1 = 1;            //TRISTATE
    TRISC0 = 1;            //TRISTATE
// --- PORTD ---
    PORTD = 0B01110000;    //SET PORT(XOE=1)
    TRISD7 = 0;            //OUT
    TRISD6 = 1;            //TRISTATE

```

```

TRISD5 = 1;           //TRISTATE
TRISD4 = 0;           //OUT
TRISD3 = 0;           //OUT
TRISD2 = 0;           //OUT
TRISD1 = 0;           //OUT
TRISD0 = 0;           //OUT
PSPMODE = 0;         //SET GENERAL I/O
// --- PORTE ---
PORTE = 0xFF;         //CLR PORT
TRISE = 0B00000111; //BIT 2-0 INPUT
ADFM = 0;             //AD Result Format Select
ADCS2 = 1;            //AD Conv. Clock Select
PCFG3 = 0;            //SET SANPLING RATE = CLK/64
PCFG2 = 0;            //
PCFG1 = 0;            //
PCFG0 = 0;            //
//-----
I2CInit();            //EEPROM(i2c bus) device init.
//-----
// Rs232Init(0x02);   //--- USART & interrupt init.
OUT_LEDS(0x00);       //表示用LED (D2~5) に表示
}
/*=====
表示用LED (D2~5) にCH値を表示
=====*/
void OUT_LEDS(unsigned char data) //表示用LED (D2~5) にCH値を表示
{
    RC5 = ( data & 0x01 );
    RD3 = (( data & 0x02 ) >> 1);
    RD7 = (( data & 0x04 ) >> 2);
}

```

List. 1 - 2 "Delay.h" program module
 • Printed-Circuit Board : LED current control
 • IC No : U4 (PIC18F452-I/P)

```

/*
(開発環境)
- MPLAB Integrated Development Environment , (Microchp Co., Ltd)
- HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
*
*      DelayUs(x)      Delay specified number of microseconds
*      DelayMs(x)     Delay specified number of milliseconds
*/
#define   MHZ      *1
#define   XTAL_FREQ  20MHZ      //命令=0.2us
#define   uS_CNT    199
#ifndef   XTAL_FREQ
#define   XTAL_FREQ  4MHZ      /* Crystal frequency in MHz */
#endif
#if      XTAL_FREQ < 8MHZ
#define   uS_CNT    238      /* 4x to make 1 mSec */
#endif
#if      XTAL_FREQ == 8MHZ
#ifndef   uS_CNT
#define   uS_CNT    244
#else
#define   uS_CNT    246
#endif
#endif
#define   FREQ_MULT  (XTAL_FREQ)/(4MHZ)
extern void DelayUs(unsigned char);
extern void DelayMs(unsigned char);
extern void DelaySec(unsigned char);

```

List. 1 - 3 "Eeprom_led.h" program module

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```

/*=====
EEPROM device. writes & reads by i2c program
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
=====*/

#ifndef EepromH
#define EepromH
#define PIC18
#if defined(PIC18)
#include <pic18.h>
#else
#include <pic.h>
#endif
#include "delay.h"
typedef unsigned char    byte;
#ifndef UnDataH
typedef unsigned short   word;    //for c++
typedef short int        sint;
typedef unsigned long    dword;
#else
typedef unsigned int     word;    //for pic c
typedef signed int       sint;
typedef unsigned long    dword;  //HIT-TECH PICC-18 manual.pdf 80p
#endif
extern byte I2CDeviceCode;        //親プログラムにて定義
extern word debugadr;
#define I2C_I0_U8    0x04        // I/O EXPANDER U8  0010 0100
#define I2C_I0_U9    0x05        // I/O EXPANDER U9  0010 0101
#define I2CEPPIC    0x10        //pic MEMORY dummy
#define I2CROM10    0xA0        //No1 EEPROM page 0 64k byte    LED基板のPage0
#define I2CROM11    0xA2        //No1 EEPROM page 1 64k byte    LED基板のPage1
#define I2CROM20    0xA4        //No2 EEPROM page 0 64k byte
#define I2CROM21    0xA6        //No2 EEPROM page 1 64k byte
#define I2CSRAM1    0xA8        //No1 SRAM 256 byte
#define I2CSRAM2    0xAA        //No2 SRAM 256 byte
#define LED_ROM_P0    0xA0        //LED基板のEEPROM(U6) Page0
#define LED_ROM_P1    0xA2        //LED基板のEEPROM(U6) Page1
#define WRITE        (I2CDeviceCode)
#define READ        (I2CDeviceCode  0x01)
#ifndef PIC18
void STARTedtn(void);
void RepeatedSTARTedtn(void);
void STOPedtn(void);

```



```

char txbyte(byte data);
void rxbyte(byte *data);
void rxLastbyte(byte *data);
#else
#define STARTcdtn()      {DelayUs(8); SEN = 1;   while(SEN == 1); }
#define RepeatedSTARTcdtn() {DelayUs(8); RSEN = 1;   while(RSEN == 1); }
#define STOPcdtn()      {DelayUs(8); PEN = 1;   while(PEN == 1); SSPIF=0; DelayUs(20);}
#define txbyte(data)    {DelayUs(6); while(STAT_BF); SSPBUF = data;   SSPIF=0; while(STAT_RW == 1); SSPIF=0;}
#define rxLastbyte(data) {DelayUs(6); RCEN = 1;   while(RCEN == 1);   Y
                        ACKDT = 1;   ACKEN = 1;   while(ACKEN == 1); data = SSPBUF;}
#endif
extern void I2CInit();
//--- EEPROM AT24C1024-----
extern byte I2CGetByte(word address);
extern char I2CPutByte(word address, byte data);
extern word I2CGetWord(word address);
extern char I2CPutWord(word address, word data);
//--- SRAM PCF8570
extern void SRMInit(void);
extern byte SRMGetByte(byte address);
extern char SRMPutByte(byte address, byte data);
extern word SRMGetWord(byte address);
extern char SRMPutWord(byte address, word data);
//--- EEPROM in PIC -----
extern byte PICGetByte(byte address);
extern char PICPutByte(byte address, byte data);
extern word PICGetWord(byte address);
extern char PICPutWord(byte address, word data);
#endif

```

List. 1 - 4 "Rs232c.h" program module

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```
/*=====
RS232C program use USART
(開発環境)
  • MPLAB Integrated Development Environment , (Microchip Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
=====*/

#define PIC18
#ifdef PIC18
  #include <pic18.h>
#else
  #include <pic.h>
#endif
#include <string.h>
#include <ctype.h>
#include "rs232c.h"
#include "Delay.h"

void OUT_ADR(unsigned char);          //SET ADDRESS TO PALCE22V10
unsigned char RS232_BUFF[BUFF_SIZ];  //受信バッファ
unsigned char RS232_INDEX;           //インデックス (蓄積バイト数)
unsigned char RS232_FLUG;            //エラーフラグ
/*=====
Received Byte Data by RS232C with interrupt.
=====*/

void interrupt RS232_RX(void)          //割り込みで受信処理
{
  if (RCIF == 1)                      //RS232割り込みの場合
  {
    RS232_FLUG = 0;                   //エラーフラグクリア
    if (FERR == 1) {RS232_FLUG |= 0x01;} //エラー発生ならフラグセット
    if (OERR == 1)
    {
      RS232_FLUG = 0x02;              //エラー発生ならフラグセット
      CREN = 0;                       //OERRをクリアする
      CREN = 1;                       //OERRをクリアする
    }
    if (RS232_INDEX < BUFF_SIZ)       //インデックスが許容範囲内なら以下を実行。
    {
      RS232_BUFF[RS232_INDEX] = RCREG; //受信データをバッファに記録する。(自動的にRCIF=0になる)
      RS232_INDEX++;                  //インデックスを更新する。
    }
  }
  else
  {
    RS232_INDEX = 0xFF;               //インデックスが許容範囲外なら、バッファフルをセット。
    RS232_BUFF[BUFF_SIZ-1] = RCREG;   //ひとまず、最新データをバッファの末尾に記録する。
  }
}
```

```

    }
}

/*=====
Init. USART
=====*/

void Rs232Init(byte frg)
{
    TRISC6 = 0;           //TX output (Send)
    TRISC7 = 1;           //RX input (Received)
    RCSTA = 0x00;         //CREN = 0           ref. P106
    SPBRG = 32;           //9600bps (FOSC=20Mhz) ref. P107
    if(frg & 0x01) TXIE = 1; //interrupt switch
    if(frg & 0x02) RCIE = 1; //interrupt switch
    TXSTA = 0x60;         //TX9=1, TXEN = 1, BRGH = 0
    RCSTA = 0xD0;         //RX9=1, SPEN = 1, CREN = 1
    RS232_INDEX = 0;     //受信用変数初期化
    RCIE = 1;             //RS232割り込みイネーブル
    RCIP = 1;             //RS232高位割り込みイネーブル
    PEIE = 1;             //周辺割り込みイネーブル
    GIE = 1;              //全体割り込みイネーブル
}

/*=====
Send Byte Data by RS232C Interface
=====*/

char Rs232PutByte(byte data)
{
    while(TRMT == 0); //TRMT == 0 buff full
    TXREG = data;
    DelayUs(250); //
    DelayUs(250); //
    return 0;
}

/*=====
Received Byte Data by RS232C Interface
=====*/

char Rs232GetByte(byte *data)
{
    char ret = 0;
    while(RCIF == 0); //RCIF == 0           ref. P23
    if(FERR == 1) ret |= 0x01;
    if(OERR == 1) ret |= 0x02;
    *data = RCREG;
    return ret;
}

/*=====
Send Word Data by RS232C Interface
=====*/

char Rs232PutWord(word data)
{
    Rs232PutByte((byte) data);
    Rs232PutByte((byte) (data >> 8));
    return 0;
}

```

```
/*=====
Received Word Data by RS232C Interface
=====*/
char Rs232GetWord(word *data)
{
    char ret;
    char dtt;          //data temp.
    ret = 0;
    ret |= Rs232GetByte(&dtt);    *data = dtt;
    ret |= Rs232GetByte(&dtt);    *data |= dtt << 8;
    return ret;
}
```

List. 1 - 5 "Command.h" program module

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```

/*****
Rs232c <-> EEPROM Access Comad
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
*****/

#ifndef byte
    typedef unsigned char    byte;
#endif
#ifndef word
    typedef unsigned int     word;
#endif
#define CMDRE    0x01    //rs232c <- EEPROM
#define CMDWR    0x02    //rs232c -> EEPROM
#define CMDREB   0x03    //rs232c <- EEPROM ブロック転送読み込み
#define  Cmdbuffsize 5    //strcut size
typedef struct {
    byte  cmd:        //command no
    byte  dev:        //device no
    word  address:    //rom address
    byte  data;
} sCMD;
typedef struct {
    byte  btPtr : 7;
    byte  btBusy : 1;
} sCmdStat;
extern sCmdStat sCmdSt;
extern void ComdInit(void);
extern void GetCommand(void);
extern void ExeCommand(void);

```

List. 1 - 6 "Time.h" program module

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```
/*=====
DS1307 device. writes & reads by i2c program
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
=====*/

#ifndef TimeH
#define TimeH
#include "eeprom.h"
#define I2CTIME      0xD0      //DALLAS Date Time Device
#define DSSEC       0x00
#define DSMIN       0x01
#define DSHOURS     0x02
#define DSDAY       0x03
#define DSDATE      0x04
#define DSMONTH     0x05
#define DSYEAR      0x06
#define DSCONTROL   0x07
extern void DS1307Init(void);
extern byte DSGetByte(word address);
extern char DSPutByte(word address, byte data);
extern void DSStart(void);
extern void DSStop(void);
extern byte DSGetSec(void);
extern byte DSGetMin(void);
extern byte DSGetHours(void);
extern byte DSGetDate(void);
extern byte DSGetMonth(void);
extern byte DSGetYear(void);
extern void DSPutSec(char sec);
extern void DSPutMin(char min);
extern void DSPutHours(char Hours);
extern void DSPutDate(char date);
extern void DSPutMonth(char month);
extern void DSPutYear(char year);
typedef struct {
    byte  bYear;          //20?? + Year (0 - 99)
    byte  bMonth;        //1 - 12
    byte  bDay;          //1 - 31
    byte  bHours;        //0 - 23
    byte  bMin;          //0 - 59
    byte  bSec;          //0 - 59
} sTime;
#endif
```

List. 1 - 7 "LED_PNL.h" program module

- Printed-Circuit Board : LED current control
- IC No : U4 (PIC18F452-I/P)

```

/*=====
CPU port assign
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
/*=====*/

/*
Port assigne
PORT-A:
  BIT7:  :Invalid
  BIT6: IN:no use
  BIT5: IN:Volt of LED CH4
  BIT4: IN:RS485_XREQ
  BIT3: IN:Volt of LED CH3
  BIT2: IN:Volt of LED CH2
  BIT1: IN:Volt of LED CH1
  BIT0: IN:Volt of LED CH0
PORT-B:
  BIT7:OUT:no use
  BIT6:OUT:no use
  BIT5:OUT:no use
  BIT4:OUT:no use
  BIT3:OUT:no use
  BIT2:OUT:no use
  BIT1:OUT:no use
  BIT0:OUT:no use
PORT-C:
  BIT7: IN:CH0_RXD
  BIT6: IN:CH1_TXD
  BIT5: IN:CH1_EN_TX
  BIT4: IN:I2C_SDA
  BIT3: IN:I2C_SCL
  BIT2: IN:XEN
  BIT1: IN:RS485_XACK
  BIT0: IN:DA_DATA_OUT
PORT-D:
  BIT7:OUT:DEBUG_LED_D5
  BIT6:  :no use
  BIT5:OUT:DIP_SW4
  BIT4:OUT:X_OE
  BIT3:OUT:RS485_REQ
  BIT2:OUT:DA_LOAD_DATA
  BIT1:OUT:DA_DATA_IN
  BIT0:OUT:DA_CLK

```

```

PORT-E:
    BIT7: :Invalid
    BIT6: :Invalid
    BIT5: :Invalid
    BIT4: :Invalid
    BIT3: :Invalid
    BIT2: IN:CH1_RXD
    BIT1: :no use
    BIT0: IN:Volt of LED CH5
    PORTD_bit1:OUT:DI
    PORTC_bit0:IN :DO
    PORTD_bit0:OUT:CLK
    PORTD_bit2:OUT:LD
*/
//----- D/A CONVERTER -----
#define DA_DI LD1
#define DA_DO RC0
#define DA_CLK LD0
#define DA_LD LD2
//----その他----
#define XOE RD4
#define FLG_EEPROM 0x00 //EEPROM用フラグ (0:通常、1:書き込み中)
#define CHK_EEPROM0 0x01 //EEPROMデータの有効性の検査用
#define CHK_EEPROM1 0x02 //EEPROMデータの有効性の検査用
#define CHK_EEPROM2 0x03 //EEPROMデータの有効性の検査用
#define TBL_U8_EEPROM 0x08 //EEPROM上で、U 8へ書くデータのアドレス
#define TBL_U9_EEPROM 0x09 //EEPROM上で、U 9へ書くデータのアドレス
#define TBL_TOP_EEPROM 0x0A //EEPROM上で、データテーブルアドレスの先頭

```


List. 1 - 8 "Main" program
 • Printed-Circuit Board : CPU
 • IC No : U1 (PIC18F452-I/P)

```

/*****
CPU基板 U1(PIC18F452)
(開発環境)
  • MPLAB Integrated Development Environment , (Microchp Co., Ltd)
  • HI-TECH C Compiler for PIC18 MCU , (Hi-tech Co., Ltd)
*****/
#define PIC18
#include <pic18.h>
  _CONFIG(1, OSCSDIS & ECRA6);           //クロックセレクト無効、OSC2はRA6として使用。
  _CONFIG(2, WDTDIS & PWRTEN & BOREN & BORV45); //WDT=無効、パワータイマー有効、BOR有効 (4.5V)
  _CONFIG(3, CCP2RC1);                 //CCP2 i/o multiplexed with RC1
  _CONFIG(4, DEBUGDIS & LVPDIS & STVREN); //デバッグ無効、低電圧書き込み有効、スタックリセット有効

#include <string.h>
#include "Delay.h"
#include "Eeprom.h"
#include "Rs232c.h"
#include "CNT_SYS.h"
static unsigned char S_ARY6A[6], S_ARY6B[6], S_ARY6C[6];
static unsigned char ZTCLK[6][6], ZTDLY[6][6], ZTSTP[6][6];
static unsigned char ZDEB_CNT;
void ZOUT_7SEG(unsigned int data);
/*=====
I2CバスをCLOSEする。
=====*/
void I2C_CLOSE(void)
{
    DelayUs(100);           //待つ
    RCO = 0;               //BUSY-OUTを無効にする。
    DelayUs(100);         //待つ
}
/*=====
I2CバスをOPENする。
=====*/
void I2C_OPEN(void)
{
    DelayUs(100);           //100usec待つ
    for ( ; ; ){           //OPENが成功するまでloopする。
        for ( ; !RC5 ; ){ //BUSY-INが0 (I2Cがフリー) になるまで、待つ。
            RCO = 1;      //BUSY-OUTを立てる
            DelayUs(100); //100usec待つ
            if (0==RC5) {break;} //BUSY-INが0なら、OPEN完了。
            //衝突した場合
            RCO = 0;      //BUSY-OUTを無効にする。
            DelayUs(4);   //4usec待つ
        }
    }
}

```

```

    } //リトライする。
    for ( ; ; ){ //SSPSTATのP=1又はS=0まで待つ。
        if (0==S) {break;} //if 'START' not detect then break.
        if (0!=P) {break;} //if 'STOP' detected then break.
    }
    for ( ; 0==RC4 ; ){ //SDA=HIまで待つ。
    for ( ; 0==RC3 ; ){ //SCL=HIまで待つ。
    DelayUs(50); //usec待つ
}
}
/*=====
6キャラクターの値を、配列変数に代入する。
=====*/
void SET2ARY (CHR5, CHR4, CHR3, CHR2, CHR1, CHR0, ARY) //SET CHRx6 TO ARREY[6] IN:CHR5, CHR4, CHR3, CHR2, CHR1, CHR0, *ARY[6]
    unsigned char CHR5, CHR4, CHR3, CHR2, CHR1, CHR0;
    unsigned char ARY[6];
{
    ARY[0] = CHR0;
    ARY[1] = CHR1;
    ARY[2] = CHR2;
    ARY[3] = CHR3;
    ARY[4] = CHR4;
    ARY[5] = CHR5;
}
/*=====
WRITE ADDRESS TO PORT B
=====*/
void OUT_ADR(unsigned char P_ADR) //SET ADDRESS TO PALCE22V10
{
    PORTB = ((P_ADR << 1) | 0x81); //MASK "INT" AND "PGD"
}
/*=====
WRITE DATA TO PORT
=====*/
void OUT_PORT(P_ADR, OUT_DATA) //WRITE TO PORT, IN:PORT ADDRESS, DATA, OUT:NONE
    unsigned char P_ADR, OUT_DATA;
{
    PORTD = OUT_DATA; //SET DATA TO DATA-BUS
    DelayUs(1);
    OUT_ADR(P_ADR); //ENABLE XCS
    DelayUs(1);
    OUT_ADR(0xFF); //DISENABLE XCS
    DelayUs(1);
}
/*=====
WRITE DATA TO REG OF FPGA
=====*/
void OUT_FPGA(FPGA_ADR, FPGA_DATA) //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
    unsigned char FPGA_ADR, FPGA_DATA;
{
    OUT_PORT(0x04, FPGA_ADR); //WRITE TO REGISTER OF FPGA_ADR
    OUT_PORT(0x03, FPGA_DATA); //WRITE TO REGISTER OF FPGA_DARA
    OUT_PORT(0x02, 0x00); //WRITE TO TARGET REGISTER OF FPGA
}
/*=====

```

配列変数 (6 バイト) を、FPGA のターゲットレジスタに書き込む。

```
=====*/
void OUT_FPGA_ARY(FPGA_ADR, ARY)          //WRITE ARREY DATA TO FPGA REG, IN:FPGA_ADR,*ARY[6]
{
    unsigned char FPGA_ADR;
    unsigned char ARY[6];
    {
        OUT_FPGA(FPGA_ADR + 0, ARY[0]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
        OUT_FPGA(FPGA_ADR + 1, ARY[1]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
        OUT_FPGA(FPGA_ADR + 2, ARY[2]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
        OUT_FPGA(FPGA_ADR + 3, ARY[3]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
        OUT_FPGA(FPGA_ADR + 4, ARY[4]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
        OUT_FPGA(FPGA_ADR + 5, ARY[5]);    //WRITE TO FPGA, IN:FPGA ADDRESS, DATA, OUT:NONE
    }
}
=====*/
Init Device
=====*/
void DeviceInit(void)
{
    SCS = 0;          //System Clock Switch (SELECT CLK PIN)
// --- RCON ---
    IPEN = 1;        //Interrupt Priority Enable(RCON-BIT7)
// --- INTCON ---
    GIEH = 0;        //HIGH Interrupt DIS Enable
    GIEL = 0;        //LOW Interrupt DIS Enable
// --- INTCON2 ---
    RBPU = 1;        //Port B Pull-Up Enable
// --- PIE1 ---
    PSP1E = 0;       //Parallel Slave Port Read/Write Interrupt
    AD1E = 0;        //AD Converter Interrupt
    RC1E = 0;        //USART Recieve Interrupt
    TX1E = 0;        //USART Transmit Interrupt
    SSP1E = 0;       //Master Synchronous Serial Port Interrupt
    CCP11E = 0;      //CCP1 Interrupt
    TMR21E = 0;      //TMR2 Overflow Interrupt
    TMR11E = 0;      //TMR1 Overflow Interrupt
// --- PIE2 ---
    EE1E = 0;        //EEPROM/FLASH Write Interrupt
    BCL1E = 0;       //Bus Collision Interrupt
    LVD1E = 0;       //Low Voltage Detect
    TMR31E = 0;      //TMR3 Overflow Interrupt
    CCP21E = 0;      //CCP2 Interrupt
// --- PORTA --- (SET TRISTATE)
    PORTA = 0xFF;    //CLR PORT A
    TRISA6 = 1;      //TRISTATE
    TRISA5 = 1;      //TRISTATE
    TRISA4 = 0;      //OUT(XOE)
    TRISA3 = 1;      //TRISTATE
    TRISA2 = 1;      //TRISTATE
    TRISA1 = 1;      //TRISTATE
    TRISA0 = 1;      //TRISTATE
// AD CONVERTER Control bits (SET ALL BIT TO PORT, A/D OFF)
    ADON = 0;        //AD CONVERTER OFF
    PCFG3 = 0;
    PCFG2 = 1;
}
```

```

PCFG1 = 1;
PCFG0 = 0;
// --- PORTB --- (SET OUT)
PORTB = 0xFF;           //SET ADR=HIGH
TRISB7 = 1;            //TRISTATE (PGD)
TRISB6 = 0;            //OUT
TRISB5 = 0;            //OUT
TRISB4 = 0;            //OUT
TRISB3 = 0;            //OUT
TRISB2 = 0;            //OUT
TRISB1 = 0;            //OUT
TRISB0 = 1;            //TRISTATE (INT)
// --- PORTC --- (SET INPUT)
PORTC = 0x00;          //CLR PORT
TRISC7 = 1;            //TRISTATE
TRISC6 = 0;            //OUT (TX1)
TRISC5 = 1;            //TRISTATE
TRISC4 = 1;            //TRISTATE
TRISC3 = 1;            //TRISTATE
TRISC2 = 0;            //OUT (PWM1)
TRISC1 = 0;            //OUT (FAN1)
TRISC0 = 0;            //OUT (BUSY-OUT)
// --- PORTD ---
PORTD = 0x00;          //SET DB=0
TRISD = 0x00;          //OUT
PSPMODE = 0;           //SET PORT-D TO GENERAL I/O
// --- PORTE ---
PORTE = 0xFF;          //CLR PORT
TRISE = 0b00000011;   //BIT1=INPUT, BIT2=OUT

I2CInit();             //EEPROM(i2c bus) device init.
}
/*=====
7セグLEDポートにデータを出力する。
=====*/
void ZOUT_7SEG(unsigned int data)           //7セグLEDに表示する。
{
    OUT_PORT(0, data);                       //7セグへ出す。
    DelayMs(100);                             //ダミーWAIT
    // DelaySec(1);                             //ダミーWAIT
}
/*=====
7セグLEDポートにデータを出力する。(WAIT有り)
=====*/
void ZOUT_7SEG_WAIT1(unsigned int data)     //7セグLEDに表示する。
{
    ZOUT_7SEG(0xAA);                          //7セグLEDに表示する。
    DelaySec(1);                               //ダミーWAIT
    ZOUT_7SEG(data);                          //7セグLEDに表示する。
    DelaySec(2);                               //ダミーWAIT
}
/*=====
7セグLEDポートにデータを出力する。(WAIT有り)
=====*/

```

```

void ZOUT_7SEG_WAIT2(unsigned int data)          //7セグLEDに表示する。
{
    ZOUT_7SEG(data);                            //7セグLEDに表示する。
    DelaySec(2);                                //ダミーWAIT
}
/*=====
7セグLEDにカウント値を表示し、カウントアップする。
=====*/
void ZDEB_CNT_UP(void)                          //7セグLEDにカウント値を表示し、カウントアップする。
{
    OUT_PORT(0, ZDEB_CNT);                      //7セグへ出す。
    DelaySec(1);                                //ダミーWAIT
    ZDEB_CNT++;
}
/*=====
main loop
=====*/
main(void)
{
    unsigned char    i, j, k;
    unsigned char    ZPOWER[5], ZON_PTN[6], ZON_FLG, ZTBL;
    unsigned char    ZPOWER_RETU_L[5][6];
    unsigned char    ZPOWER_RETU_H[5][6];
//変数の初期化
    ZDEB_CNT = 0;                                //デバッグ用カウンタ
    DeviceInit();                                //I/Oポート初期化
    DelayMs(250);                                //ダミーWAIT
    I2C_CLOSE();                                 //I2Cバスを閉じる（開放する）
//-----< CPU2からの指示を読み、ハードを設定 >-----
    ZOUT_7SEG(0x10);                             //7セグLEDに表示する。
    I2C_OPEN();                                  //I2Cバスを占有する。
    DelaySec(1);                                 //ダミーWAIT（秒）
    ZOUT_7SEG(0x11);                             //7セグLEDに表示する。
    I2CDeviceCode = I2CSRAM1;                   //SRAM(No1) (A8)
    i = SRMGetByte(0x0);
    ZOUT_7SEG(0x12);                             //7セグLEDに表示する。
    ZOUT_7SEG(0x13);                             //7セグLEDに表示する。
    I2C_CLOSE();                                 //I2Cバスを閉じる（開放する）
    ZOUT_7SEG(0x14);                             //7セグLEDに表示する。
    for(;;){                                     //LOOP1:
        ZOUT_7SEG(0xFE);                         //7セグLEDに表示する。
        DelaySec(3);                             //ダミーWAIT（秒）
        I2C_OPEN();                              //I2Cバスを占有する。
        I2CDeviceCode = I2CSRAM1;               //SRAM(No1) (A8)
        if (0 != SRMGetByte(0x50)){              //SRAM(50H)更新フラグが、更新状態なら、以下を実行。
            ZOUT_7SEG(0x55);                     //7セグLEDに表示する。
            for (i=0; i!=5; i++){                //SRAMのテーブル情報（03H-12H）をPOWER[5列分]に代入する。
                ZPOWER[i] = SRMGetByte(0x03+i);
            }
            for (i=0; i!=6; i++){                //SRAMのテーブル情報（13H-1AH）をON-PTN[8ch分]に代入する。
                ZON_PTN[i] = SRMGetByte(0x13+i);
            }
            ZON_FLG = SRMGetByte(0x21);          //SRAMのテーブル情報（21H）をON-FLGに代入する。
            for (i=0; i!=6; i++){                //CH0から5まで6チャンネル分

```

```

ZTBL=ZON_PTN[i]; //ON-PTN[CH]を読み、TBL番号に代入する。
for (j=0; j!=6; j++){ //連続する、6バイトを代入する。
    I2CDeviceCode = I2CROM11; //EEPROM(No1) page1(A2)
    ZTCLK[i][j] = I2CGetByte(ZEEP_TBL_TC+ZTBL*64+j); //EEPROMから基本周期を読み代入。
    ZTDLY[i][j] = I2CGetByte(ZEEP_TBL_TD+ZTBL*64+j); //EEPROMから基本遅延を読み代入。
    ZTSTP[i][j] = I2CGetByte(ZEEP_TBL_TS+ZTBL*64+j); //EEPROMから基本消灯を読み代入。
}
}

for (i=0; i!=5; i++){ //0から4まで、5列分。
    ZTBL=ZPOWER[i]; //POWER[RETU]を読み、TBL番号に代入する。
    for (j=0; j!=6; j++){ //6CH分を代入する。
        I2CDeviceCode = I2CROM11; //EEPROM(No1) page1(A2)
        ZPOWER_RETU_L[i][j] = I2CGetByte(ZEEP_TBL_PWR+ZTBL*32+j*2-0); //EEPROMから電流値を読み代入。
        ZPOWER_RETU_H[i][j] = I2CGetByte(ZEEP_TBL_PWR+ZTBL*32+j*2+1); //EEPROMから電流値を読み代入。
    }
}

I2CDeviceCode = I2CSRAM1; //SRAM(No1) (A8)
for ( ; ! SRMPutByte(0X50,0x00); ){ //SRAM(50H)に、00Hを書き込む。(読み完了を通知)
    I2C_CLOSE(); //I2Cバスを閉じる(開放する)
    RA4 = 1; //XOE=1、全LEDを消灯する。
    OUT_FPGA(0xE0,0x00); //FPGA内LEDの点灯フラグを0(消灯)にする。
    //6CH分をFPGAへ転送。
    OUT_FPGA_ARY(FPGA_ADR_CNT_MAX,&ZTCLK[0][0]); //WRITE ARREY DATA TO FPGA

    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN0,&ZTDLY[0][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX0,&ZTSTP[0][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN1,&ZTDLY[1][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX1,&ZTSTP[1][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN2,&ZTDLY[2][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX2,&ZTSTP[2][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN3,&ZTDLY[3][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX3,&ZTSTP[3][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN4,&ZTDLY[4][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX4,&ZTSTP[4][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MIN5,&ZTDLY[5][0]); //WRITE ARREY DATA TO FPGA
    OUT_FPGA_ARY(FPGA_ADR_WIN_MAX5,&ZTSTP[5][0]); //WRITE ARREY DATA TO FPGA
    //FOR(RETU=0,RETU<=4,++)、5列分を各DIG基板に設定
    //DIG基板にCH毎の電流値PWR-RETU[RETU][j]を転送。
    OUT_FPGA(0xE0,ZON_FLG); //FPGA内LEDの点灯フラグに書き込む。
    RA4 = 0; //XOE、全LEDを有効にする。
}
I2C_CLOSE(); //I2Cバスを閉じる(開放する)
} //LOOPする。
} //END OF MAIN

```

List. 1 - 9 Hardware Description Language of FPGA
 • Printed-Circuit Board : FPGA
 • IC No : U1 (EPM2201F256C5)

```

TITLE "FPGA-EPM2201F256";
SUBDESIGN FPGA-EPM2201F256
(
-- (開発環境)
--   • Quartus II Development Software , (Altera Co., Ltd)
-----< DEFINE I/O ASSIGN >-----
D[7..0]      : INPUT;
CLR          : INPUT;
OE           : INPUT;
CLK0         : INPUT;
CLK1         : INPUT;
CLK2         : INPUT;
CLK3         : INPUT;
XCS[5..2]    : INPUT;
AD[5..4]     : INPUT;
XRES         : INPUT;
XOER         : INPUT;
DEB[15..0]   : OUTPUT;
CH[7..0]     : OUTPUT;
CLK1_OUT     : OUTPUT;
CLK2_OUT     : OUTPUT;
)
VARIABLE
CNT[47..0]   : DFF;
CNT_MAX[47..0] : LATCH;
WIN_MIN0[47..0] : LATCH;
WIN_MIN1[47..0] : LATCH;
WIN_MIN2[47..0] : LATCH;
WIN_MIN3[47..0] : LATCH;
WIN_MIN4[47..0] : LATCH;
WIN_MIN5[47..0] : LATCH;
WIN_MAX0[47..0] : LATCH;
WIN_MAX1[47..0] : LATCH;
WIN_MAX2[47..0] : LATCH;
WIN_MAX3[47..0] : LATCH;
WIN_MAX4[47..0] : LATCH;
WIN_MAX5[47..0] : LATCH;
CH_EN[7..0]    : LATCH;      --各CHイネーブル用レジスター (0 : 消灯、1 : 点灯)
XCS_R[5..2]    : DFF;        --非同期信号なので、ラッチする。
CNT_CLK1[5..0] : DFF;
CH_R[7..0]     : DFF;        --点灯信号用レジスター
ADR_R[7..0]    : LATCH;
DATA_R[7..0]   : LATCH;
-----< DEFAULTS >-----

```

```

BEGIN
DEFAULTS
    CNT[47..0].D      = GND;
    CNT_MAX[47..0].D  = GND;
    WIN_MIN0[47..0].D = GND;
    WIN_MIN1[47..0].D = GND;
    WIN_MIN2[47..0].D = GND;
    WIN_MIN3[47..0].D = GND;
    WIN_MIN4[47..0].D = GND;
    WIN_MIN5[47..0].D = GND;
    WIN_MAX0[47..0].D = GND;
    WIN_MAX1[47..0].D = GND;
    WIN_MAX2[47..0].D = GND;
    WIN_MAX3[47..0].D = GND;
    WIN_MAX4[47..0].D = GND;
    WIN_MAX5[47..0].D = GND;
    XCS_R[5..2].D     = GND;
    CH_R[7..0].D      = GND;
    CH_EN[7..0].D     = GND;
    CNT_CLK1[5..0].D  = GND;
    ADR_R[7..0].D     = GND;
    DATA_R[7..0].D   = GND;
    CLK1_OUT          = GND;
    CLK2_OUT          = GND;
END      DEFAULTS;
-----< ASSIGN CLK >-----
    CNT[47..0].CLK    = CLK0;
    CNT_CLK1[5..0].CLK = CLK0;
    XCS_R[5..2].CLK   = CLK1;      --低速CLKで叩き、安定化する。
    CH_R[7..0].CLK    = CLK0;
-----< ASSIGN ENABLE OF LATCH >-----
    XCS_R[5..2].D     = XCS[5..2];
    ADR_R[7..0].ENA   = !XCS_R[4];
    ADR_R[7..0].D     = D[7..0];
    DATA_R[7..0].ENA = !XCS_R[3];
    DATA_R[7..0].D   = D[7..0];
CASE ADR_R[7..0] IS
----E0
    WHEN H"E0" =>
        CH_EN[7..0].ENA = !XCS_R[2];
----F0
    WHEN H"F0" =>
        CNT_MAX[7..0].ENA = !XCS_R[2];
    WHEN H"F1" =>
        CNT_MAX[15..8].ENA = !XCS_R[2];
    WHEN H"F2" =>
        CNT_MAX[23..16].ENA = !XCS_R[2];
    WHEN H"F3" =>
        CNT_MAX[31..24].ENA = !XCS_R[2];
    WHEN H"F4" =>
        CNT_MAX[39..32].ENA = !XCS_R[2];
    WHEN H"F5" =>
        CNT_MAX[47..40].ENA = !XCS_R[2];
----0

```



```

WHEN H"00" =>
    WIN_MIN0[7..0].ENA = !XCS_R[2];
WHEN H"01" =>
    WIN_MIN0[15..8].ENA = !XCS_R[2];
WHEN H"02" =>
    WIN_MIN0[23..16].ENA = !XCS_R[2];
WHEN H"03" =>
    WIN_MIN0[31..24].ENA = !XCS_R[2];
WHEN H"04" =>
    WIN_MIN0[39..32].ENA = !XCS_R[2];
WHEN H"05" =>
    WIN_MIN0[47..40].ENA = !XCS_R[2];
WHEN H"08" =>
    WIN_MAX0[7..0].ENA = !XCS_R[2];
WHEN H"09" =>
    WIN_MAX0[15..8].ENA = !XCS_R[2];
WHEN H"0A" =>
    WIN_MAX0[23..16].ENA = !XCS_R[2];
WHEN H"0B" =>
    WIN_MAX0[31..24].ENA = !XCS_R[2];
WHEN H"0C" =>
    WIN_MAX0[39..32].ENA = !XCS_R[2];
WHEN H"0D" =>
    WIN_MAX0[47..40].ENA = !XCS_R[2];
----1
WHEN H"10" =>
    WIN_MIN1[7..0].ENA = !XCS_R[2];
WHEN H"11" =>
    WIN_MIN1[15..8].ENA = !XCS_R[2];
WHEN H"12" =>
    WIN_MIN1[23..16].ENA = !XCS_R[2];
WHEN H"13" =>
    WIN_MIN1[31..24].ENA = !XCS_R[2];
WHEN H"14" =>
    WIN_MIN1[39..32].ENA = !XCS_R[2];
WHEN H"15" =>
    WIN_MIN1[47..40].ENA = !XCS_R[2];
WHEN H"18" =>
    WIN_MAX1[7..0].ENA = !XCS_R[2];
WHEN H"19" =>
    WIN_MAX1[15..8].ENA = !XCS_R[2];
WHEN H"1A" =>
    WIN_MAX1[23..16].ENA = !XCS_R[2];
WHEN H"1B" =>
    WIN_MAX1[31..24].ENA = !XCS_R[2];
WHEN H"1C" =>
    WIN_MAX1[39..32].ENA = !XCS_R[2];
WHEN H"1D" =>
    WIN_MAX1[47..40].ENA = !XCS_R[2];
----2
WHEN H"20" =>
    WIN_MIN2[7..0].ENA = !XCS_R[2];
WHEN H"21" =>
    WIN_MIN2[15..8].ENA = !XCS_R[2];

```

```

WHEN H"22" =>
    WIN_MIN2[23..16].ENA = !XCS_R[2];
WHEN H"23" =>
    WIN_MIN2[31..24].ENA = !XCS_R[2];
WHEN H"24" =>
    WIN_MIN2[39..32].ENA = !XCS_R[2];
WHEN H"25" =>
    WIN_MIN2[47..40].ENA = !XCS_R[2];
WHEN H"28" =>
    WIN_MAX2[7..0].ENA = !XCS_R[2];
WHEN H"29" =>
    WIN_MAX2[15..8].ENA = !XCS_R[2];
WHEN H"2A" =>
    WIN_MAX2[23..16].ENA = !XCS_R[2];
WHEN H"2B" =>
    WIN_MAX2[31..24].ENA = !XCS_R[2];
WHEN H"2C" =>
    WIN_MAX2[39..32].ENA = !XCS_R[2];
WHEN H"2D" =>
    WIN_MAX2[47..40].ENA = !XCS_R[2];

```

----3

```

WHEN H"30" =>
    WIN_MIN3[7..0].ENA = !XCS_R[2];
WHEN H"31" =>
    WIN_MIN3[15..8].ENA = !XCS_R[2];
WHEN H"32" =>
    WIN_MIN3[23..16].ENA = !XCS_R[2];
WHEN H"33" =>
    WIN_MIN3[31..24].ENA = !XCS_R[2];
WHEN H"34" =>
    WIN_MIN3[39..32].ENA = !XCS_R[2];
WHEN H"35" =>
    WIN_MIN3[47..40].ENA = !XCS_R[2];
WHEN H"38" =>
    WIN_MAX3[7..0].ENA = !XCS_R[2];
WHEN H"39" =>
    WIN_MAX3[15..8].ENA = !XCS_R[2];
WHEN H"3A" =>
    WIN_MAX3[23..16].ENA = !XCS_R[2];
WHEN H"3B" =>
    WIN_MAX3[31..24].ENA = !XCS_R[2];
WHEN H"3C" =>
    WIN_MAX3[39..32].ENA = !XCS_R[2];
WHEN H"3D" =>
    WIN_MAX3[47..40].ENA = !XCS_R[2];

```

----4

```

WHEN H"40" =>
    WIN_MIN4[7..0].ENA = !XCS_R[2];
WHEN H"41" =>
    WIN_MIN4[15..8].ENA = !XCS_R[2];
WHEN H"42" =>
    WIN_MIN4[23..16].ENA = !XCS_R[2];
WHEN H"43" =>
    WIN_MIN4[31..24].ENA = !XCS_R[2];

```

```

WHEN H*44* =>
    WIN_MIN4[39..32].ENA = !XCS_R[2];
WHEN H*45* =>
    WIN_MIN4[47..40].ENA = !XCS_R[2];
WHEN H*48* =>
    WIN_MAX4[7..0].ENA = !XCS_R[2];
WHEN H*49* =>
    WIN_MAX4[15..8].ENA = !XCS_R[2];
WHEN H*4A* =>
    WIN_MAX4[23..16].ENA = !XCS_R[2];
WHEN H*4B* =>
    WIN_MAX4[31..24].ENA = !XCS_R[2];
WHEN H*4C* =>
    WIN_MAX4[39..32].ENA = !XCS_R[2];
WHEN H*4D* =>
    WIN_MAX4[47..40].ENA = !XCS_R[2];
----5
WHEN H*50* =>
    WIN_MIN5[7..0].ENA = !XCS_R[2];
WHEN H*51* =>
    WIN_MIN5[15..8].ENA = !XCS_R[2];
WHEN H*52* =>
    WIN_MIN5[23..16].ENA = !XCS_R[2];
WHEN H*53* =>
    WIN_MIN5[31..24].ENA = !XCS_R[2];
WHEN H*54* =>
    WIN_MIN5[39..32].ENA = !XCS_R[2];
WHEN H*55* =>
    WIN_MIN5[47..40].ENA = !XCS_R[2];
WHEN H*58* =>
    WIN_MAX5[7..0].ENA = !XCS_R[2];
WHEN H*59* =>
    WIN_MAX5[15..8].ENA = !XCS_R[2];
WHEN H*5A* =>
    WIN_MAX5[23..16].ENA = !XCS_R[2];
WHEN H*5B* =>
    WIN_MAX5[31..24].ENA = !XCS_R[2];
WHEN H*5C* =>
    WIN_MAX5[39..32].ENA = !XCS_R[2];
WHEN H*5D* =>
    WIN_MAX5[47..40].ENA = !XCS_R[2];
----
WHEN OTHERS =>
    CNT_MAX[47..0].ENA = GND;
    CH_EN[7..0].ENA = GND;
    WIN_MIN0[47..0].ENA = GND;
    WIN_MIN1[47..0].ENA = GND;
    WIN_MIN2[47..0].ENA = GND;
    WIN_MIN3[47..0].ENA = GND;
    WIN_MIN4[47..0].ENA = GND;
    WIN_MIN5[47..0].ENA = GND;
    WIN_MAX0[47..0].ENA = GND;
    WIN_MAX1[47..0].ENA = GND;
    WIN_MAX2[47..0].ENA = GND;

```

```

        WIN_MAX3[47..0].ENA      = GND;
        WIN_MAX4[47..0].ENA      = GND;
        WIN_MAX5[47..0].ENA      = GND;
    END CASE;

    -----< LOGIC >-----
    CH_EN[7..0].D                = DATA_R[7..0].Q;

    -----
    CNT_MAX[47..40].D            = DATA_R[7..0].Q;
    CNT_MAX[39..32].D            = DATA_R[7..0].Q;
    CNT_MAX[31..24].D            = DATA_R[7..0].Q;
    CNT_MAX[23..16].D            = DATA_R[7..0].Q;
    CNT_MAX[15..8].D             = DATA_R[7..0].Q;
    CNT_MAX[7..0].D              = DATA_R[7..0].Q;

    -----
    WIN_MIN0[47..40].D           = DATA_R[7..0].Q;
    WIN_MIN0[39..32].D           = DATA_R[7..0].Q;
    WIN_MIN0[31..24].D           = DATA_R[7..0].Q;
    WIN_MIN0[23..16].D           = DATA_R[7..0].Q;
    WIN_MIN0[15..8].D            = DATA_R[7..0].Q;
    WIN_MIN0[7..0].D             = DATA_R[7..0].Q;

    -----
    WIN_MAX0[47..40].D           = DATA_R[7..0].Q;
    WIN_MAX0[39..32].D           = DATA_R[7..0].Q;
    WIN_MAX0[31..24].D           = DATA_R[7..0].Q;
    WIN_MAX0[23..16].D           = DATA_R[7..0].Q;
    WIN_MAX0[15..8].D            = DATA_R[7..0].Q;
    WIN_MAX0[7..0].D             = DATA_R[7..0].Q;

    -----
    WIN_MIN1[47..40].D           = DATA_R[7..0].Q;
    WIN_MIN1[39..32].D           = DATA_R[7..0].Q;
    WIN_MIN1[31..24].D           = DATA_R[7..0].Q;
    WIN_MIN1[23..16].D           = DATA_R[7..0].Q;
    WIN_MIN1[15..8].D            = DATA_R[7..0].Q;
    WIN_MIN1[7..0].D             = DATA_R[7..0].Q;

    -----
    WIN_MAX1[47..40].D           = DATA_R[7..0].Q;
    WIN_MAX1[39..32].D           = DATA_R[7..0].Q;
    WIN_MAX1[31..24].D           = DATA_R[7..0].Q;
    WIN_MAX1[23..16].D           = DATA_R[7..0].Q;
    WIN_MAX1[15..8].D            = DATA_R[7..0].Q;
    WIN_MAX1[7..0].D             = DATA_R[7..0].Q;

    -----
    WIN_MIN2[47..40].D           = DATA_R[7..0].Q;
    WIN_MIN2[39..32].D           = DATA_R[7..0].Q;
    WIN_MIN2[31..24].D           = DATA_R[7..0].Q;
    WIN_MIN2[23..16].D           = DATA_R[7..0].Q;
    WIN_MIN2[15..8].D            = DATA_R[7..0].Q;
    WIN_MIN2[7..0].D             = DATA_R[7..0].Q;

    -----
    WIN_MAX2[47..40].D           = DATA_R[7..0].Q;
    WIN_MAX2[39..32].D           = DATA_R[7..0].Q;
    WIN_MAX2[31..24].D           = DATA_R[7..0].Q;
    WIN_MAX2[23..16].D           = DATA_R[7..0].Q;
    WIN_MAX2[15..8].D            = DATA_R[7..0].Q;

```

```

WIN_MAX2[7..0].D = DATA_R[7..0].Q;
-----
WIN_MIN3[47..40].D = DATA_R[7..0].Q;
WIN_MIN3[39..32].D = DATA_R[7..0].Q;
WIN_MIN3[31..24].D = DATA_R[7..0].Q;
WIN_MIN3[23..16].D = DATA_R[7..0].Q;
WIN_MIN3[15..8].D = DATA_R[7..0].Q;
WIN_MIN3[7..0].D = DATA_R[7..0].Q;
-----
WIN_MAX3[47..40].D = DATA_R[7..0].Q;
WIN_MAX3[39..32].D = DATA_R[7..0].Q;
WIN_MAX3[31..24].D = DATA_R[7..0].Q;
WIN_MAX3[23..16].D = DATA_R[7..0].Q;
WIN_MAX3[15..8].D = DATA_R[7..0].Q;
WIN_MAX3[7..0].D = DATA_R[7..0].Q;
-----
WIN_MIN4[47..40].D = DATA_R[7..0].Q;
WIN_MIN4[39..32].D = DATA_R[7..0].Q;
WIN_MIN4[31..24].D = DATA_R[7..0].Q;
WIN_MIN4[23..16].D = DATA_R[7..0].Q;
WIN_MIN4[15..8].D = DATA_R[7..0].Q;
WIN_MIN4[7..0].D = DATA_R[7..0].Q;
-----
WIN_MAX4[47..40].D = DATA_R[7..0].Q;
WIN_MAX4[39..32].D = DATA_R[7..0].Q;
WIN_MAX4[31..24].D = DATA_R[7..0].Q;
WIN_MAX4[23..16].D = DATA_R[7..0].Q;
WIN_MAX4[15..8].D = DATA_R[7..0].Q;
WIN_MAX4[7..0].D = DATA_R[7..0].Q;
-----
WIN_MIN5[47..40].D = DATA_R[7..0].Q;
WIN_MIN5[39..32].D = DATA_R[7..0].Q;
WIN_MIN5[31..24].D = DATA_R[7..0].Q;
WIN_MIN5[23..16].D = DATA_R[7..0].Q;
WIN_MIN5[15..8].D = DATA_R[7..0].Q;
WIN_MIN5[7..0].D = DATA_R[7..0].Q;
-----
WIN_MAX5[47..40].D = DATA_R[7..0].Q;
WIN_MAX5[39..32].D = DATA_R[7..0].Q;
WIN_MAX5[31..24].D = DATA_R[7..0].Q;
WIN_MAX5[23..16].D = DATA_R[7..0].Q;
WIN_MAX5[15..8].D = DATA_R[7..0].Q;
WIN_MAX5[7..0].D = DATA_R[7..0].Q;
-----
< XCS取込み用低速CLK >-----
CNT_CLK1[5..0].D = CNT_CLK1[5..0].Q + 1; --XCS取込み用低速CLK作成。
-----< 主カウンター動作部分（0からMAX値までカウントする。） >-----
IF (CNT[47..0].Q >= CNT_MAX[47..0].Q) THEN --MAX値になったら、0に戻る。
    CNT[47..0].D = GND;
ELSE
    CNT[47..0].D = CNT[47..0].Q + 1; --通常は、+1する。
END IF;
-----< 点灯信号生成（CNT値が、MINからMAX値の期間のみONする。） >-----
--CHO
IF (WIN_MAX0[47..0] == WIN_MIN0[47..0]) THEN

```

```

        CH_R[0].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN0[47..0]) & (CNT[47..0].Q < WIN_MAX0[47..0])) THEN
        CH_R[0].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[0].D = GND;                --これら以外なら、GND
END IF;
--CH1
IF (WIN_MAX1[47..0] == WIN_MIN1[47..0]) THEN
        CH_R[1].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN1[47..0]) & (CNT[47..0].Q < WIN_MAX1[47..0])) THEN
        CH_R[1].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[1].D = GND;                --これら以外なら、GND
END IF;
--CH2
IF (WIN_MAX2[47..0] == WIN_MIN2[47..0]) THEN
        CH_R[2].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN2[47..0]) & (CNT[47..0].Q < WIN_MAX2[47..0])) THEN
        CH_R[2].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[2].D = GND;                --これら以外なら、GND
END IF;
--CH3
IF (WIN_MAX3[47..0] == WIN_MIN3[47..0]) THEN
        CH_R[3].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN3[47..0]) & (CNT[47..0].Q < WIN_MAX3[47..0])) THEN
        CH_R[3].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[3].D = GND;                --これら以外なら、GND
END IF;
--CH4
IF (WIN_MAX4[47..0] == WIN_MIN4[47..0]) THEN
        CH_R[4].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN4[47..0]) & (CNT[47..0].Q < WIN_MAX4[47..0])) THEN
        CH_R[4].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[4].D = GND;                --これら以外なら、GND
END IF;
--CH5
IF (WIN_MAX5[47..0] == WIN_MIN5[47..0]) THEN
        CH_R[5].D = GND;                --N I N=MAXならば、消灯する。
ELSIF ((CNT[47..0].Q >= WIN_MIN5[47..0]) & (CNT[47..0].Q < WIN_MAX5[47..0])) THEN
        CH_R[5].D = VCC;                --CNT=M I Nならば、点灯する。
ELSE
        CH_R[5].D = GND;                --これら以外なら、GND
END IF;
-----< ASSIGN OUTPUT >-----
DEB[15..0]          = CNT[15..0].Q;
CLK1_OUT            = CNT_CLK1[5].Q;      --C P Uのポートのポーリング周期用
CH[7..0]            = !(CH_R[7..0].Q & CH_EX[7..0]);  --イネーブルと&して、負論理で出す。
END;
```

List. 1 - 10 Hardware Description Language of U7
 • Printed-Circuit Board : CPU
 • IC No : U7 (PALCE22V10H-15PC/4)

```

TITLE CPU-U7.PDS
PATTERN A
REVISION 1.0
AUTHOR OHASHI
CHIP CPU-U7 PAL22V10
: (開発環境)
:   PALASM , (AMD Co., Ltd)
:----- PIN Declarations -----
PIN 1      CLOCK      COMB      : INPUT
PIN 2..7   AD[5..0]   COMB      : INPUT
PIN 8      DIR        COMB      : INPUT
PIN 12     GND
PIN 14..23 XCS[9..0]  REG      : OUTPUT
PIN 24     VCC
:----- State Segment -----
EQUATIONS
XCS[0] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[1] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[2] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[3] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[4] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[5] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[6] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[7] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[8] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
XCS[9] = /(AD[5] * AD[4] * AD[3] * AD[2] * AD[1] * AD[0] )
:----- Simulation Segment -----
SIMULATION
trace_on XCS[9..0]
SETF AD[5] AD[4] AD[3] AD[2] AD[1] AD[0]
clockf clock
clockf clock
clockf clock
clockf clock
SETF /AD[5] /AD[4] /AD[3] /AD[2] /AD[1] /AD[0]
clockf clock
clockf clock
SETF /AD[5] /AD[4] /AD[3] /AD[2] /AD[1] AD[0]
clockf clock
clockf clock
SETF /AD[5] /AD[4] /AD[3] /AD[2] AD[1] /AD[0]
clockf clock
clockf clock
SETF /AD[5] /AD[4] /AD[3] /AD[2] AD[1] AD[0]

```

```

clockf    clock
clockf    clock
SETF /AD[5] /AD[4] /AD[3] AD[2] /AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] /AD[3] AD[2] /AD[1] AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] /AD[3] AD[2] AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] /AD[3] AD[2] AD[1] AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] AD[3] /AD[2] /AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] AD[3] /AD[2] AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] AD[3] AD[2] /AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] AD[3] AD[2] AD[1] /AD[0]
clockf    clock
clockf    clock
SETF /AD[5] /AD[4] AD[3] AD[2] AD[1] AD[0]
clockf    clock
clockf    clock
SETF /AD[5] AD[4] /AD[3] /AD[2] /AD[1] /AD[0]
clockf    clock
clockf    clock
trace_off

```


付録2. LEDドライブ部の回路動作

LEDドライブ部の回路動作概要は、以下である。

①CN3 の差動信号入力 (CH0 ~ CH5) がON状態ならば該当するチャンネルのLEDを点灯し、OFF状態ならば消灯する。

②点灯時の電流値 (光量) は、CN4 の点灯時の電流値制御信号 (V-ON0 ~ 5) と電流レンジ切換信号 (I-x00mA CH0 ~ 5) の状態によって決まる。

この電流レンジ切換信号と電流値制御信号を巧みに用いた制御方式により高精度な電流制御を実現した (特許: 大橋・小松, 2007)。

③さらに、出力段にて 2 個のトランジスタをカスコード接続回路構成にしたことで、高速な点滅制御を実現した (特許: 大橋・相良, 2007)。

④さらに、出力段のトランジスタに対して並列に抵抗器を接続することで、本トランジスタへの印加電圧を軽減させた (特許: 大橋・相良, 2006)。