

研究テーマ 「ものづくりコンテスト電子回路部門への挑戦」

長野県岡谷工業高校 情報技術科 3年 塩澤 匠生  
 情報技術科 2年 有賀 啓真、花岡 良太郎、竹島 光星  
 情報技術科 1年 小松 眞丈、東 馳斗

情報技術科 竹内 一郎

1. 目的

昨年度の県大会で優勝し、北信越大会へ参加しました。今年度は、さらに上位を目指し、練習に取り組み、回路設計・製作技術・プログラミング技術を向上させることを目的とします。

2. ものづくり電子回路部門について

【競技概要】

設計仕様に基づき、支給される電子部品等を用いて回路基板設計・製作をします。

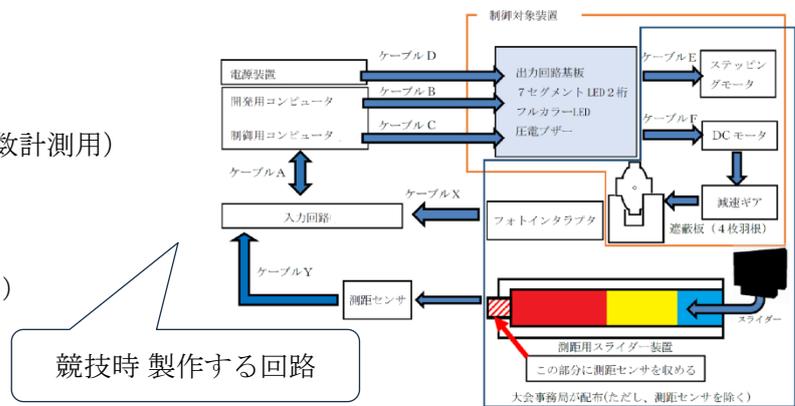
また、大会当日に提示される仕様に基づいたプログラムを作成し、制御用マイコンにプログラムを転送し実行させます。

【入力回路】

- ・タクトスイッチ、トグルスイッチ
- ・フォトインタラプタ (DCモータ回転数計測用)
- ・PSD 測距センサ

【出力回路】

- ・7セグメントLED (2個)
- ・フルカラーLED、DCモータ (1個)
- ・ステップモータ (1個)
- ・圧電ブザー (1個)



課題のシステム図 (実施要項より)

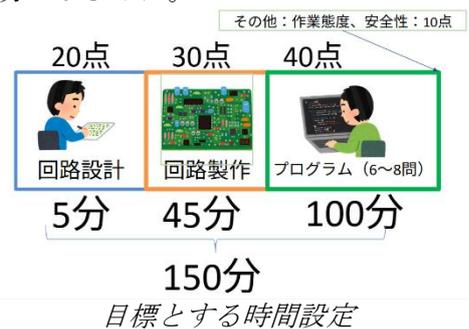
3. 研究内容

3. 1 審査基準の分析と時間配分の目標設定

審査基準は下記に示された、主にプログラミング技術と、実際に製作した入力用基板のできが採点されます。入力用基板の評価は減点法(昨年度の傾向から)であるため、練習を重ねることで減点をほとんど無くすることができるので、あまり点差がつかないことが分かりました。

その逆に、プログラミング技術では、どれだけ多くの課題を正しくクリアできたかを加点式で評価されます。また、難題であればあるほど、加点が大きくなるので、このプログラミングにかかる時間を競技の中で、どれだけ多く確保できるのかが重要だと考えました。

そこで、右図に示すように入力回路の製作時間を短くし(50分以内)し、プログラミングに充てる時間を増やすことを目標とした。



【審査基準 (実施要項より)】

| 8. 審査<br>(1) 審査対象<br>①入力回路①の図面<br>②入力回路①<br>③プログラム課題の動作状況<br>④その他(作業態度等) | 9. 採点基準<br>(1) 採点項目と観点                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                   |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|----|----|---------|----|-------------|--------|----|-----------------------------------|-----|----|--------------------------|-----|----|---------------------------------|----|-----|--|
|                                                                          | <table border="1"> <thead> <tr> <th>項目</th> <th>点数</th> <th>観点</th> </tr> </thead> <tbody> <tr> <td>プログラム動作</td> <td>40</td> <td>・完成審査での動作状況</td> </tr> <tr> <td>組み立て技術</td> <td>30</td> <td>・動作状況・部品処理(取付損傷)<br/>・ハンダの状態・配線・配置</td> </tr> <tr> <td>設計力</td> <td>20</td> <td>・図面の正確さ、完成度<br/>・配置・記号・文字</td> </tr> <tr> <td>その他</td> <td>10</td> <td>・作業態度・作業の安全性<br/>・工具及び部品の取り扱い・清掃</td> </tr> <tr> <td>合計</td> <td>100</td> <td></td> </tr> </tbody> </table> | 項目                                | 点数 | 観点 | プログラム動作 | 40 | ・完成審査での動作状況 | 組み立て技術 | 30 | ・動作状況・部品処理(取付損傷)<br>・ハンダの状態・配線・配置 | 設計力 | 20 | ・図面の正確さ、完成度<br>・配置・記号・文字 | その他 | 10 | ・作業態度・作業の安全性<br>・工具及び部品の取り扱い・清掃 | 合計 | 100 |  |
| 項目                                                                       | 点数                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 観点                                |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
| プログラム動作                                                                  | 40                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ・完成審査での動作状況                       |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
| 組み立て技術                                                                   | 30                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ・動作状況・部品処理(取付損傷)<br>・ハンダの状態・配線・配置 |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
| 設計力                                                                      | 20                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ・図面の正確さ、完成度<br>・配置・記号・文字          |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
| その他                                                                      | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ・作業態度・作業の安全性<br>・工具及び部品の取り扱い・清掃   |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |
| 合計                                                                       | 100                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                   |    |    |         |    |             |        |    |                                   |     |    |                          |     |    |                                 |    |     |  |

### 3. 2 プログラミング技術の習得

1・2年生にも理解しやすく、基本的なプログラミング技術を身に着けることを目標とし、基本的な記述方法（プログラミングの記述方針）、配布するサンプルプログラムについて考案しました。

これまで、それぞれの出力対象の制御方法を学んでから、予想される課題に取り組んできました。しかし、出力対象の制御方法の理解には時間がかかり、大会当日になっても何も動作させることができないなんてこともこれまでは多くあったそうです。そこで、今年度から難しい部分をブラックボックス化し、**最初に絶対に理解してほしい内容**と、**最初は理解しなくても動作できる内容**に分け、マイコンを使った入出力制御の基本的なプログラミングについて学習することにしました。また、1年生でも簡単な課題にチャレンジできるようになることを目標としました。

まとめたものを、下記に示します。また、これを私（3年塩澤）が実際に指導することにしました。※出力対象の制御方法については、全体のプログラムの記述方法の理解が進むにつれ、少しずつ説明することにしました。

#### <最初に理解してほしい内容と理解しなくてよい内容>

| 理解してほしい                                                                                                                                                                                                                                                              | 理解しなくてよい                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• SWICH case 文を使った工程管理<br/>(例)<br/>case :10<br/>  patten =20;<br/>  break;</li> <li>case :20<br/>  patten =30;<br/>  break;</li> <li>• 各種値と if 文を使った分岐構文</li> <li>• 変数のインクリメントやデクリメント</li> <li>• 出力対象をフラグを使って管理・制御</li> </ul> | <ul style="list-style-type: none"> <li>• 基本的に割り込み関数内</li> <li>• DC モータの回転数(フォトインタラプ 外による値カウント)の取得</li> <li>• DC モータの回転速度制御方法</li> <li>• ステッピングモータの回転制御方法</li> <li>• 圧電ブザーの鳴動の制御方法</li> <li>• 7セグメント LED のダイナミック制御方法</li> </ul> |

#### <プログラミングの記述方針（考え方）> は、最初は理解しなくてもいい部分

|       | メインループ                                                      | 割り込み(1ms 間隔)                             |                                        |
|-------|-------------------------------------------------------------|------------------------------------------|----------------------------------------|
| 工程の移行 | パターン（工程）の移行管理<br>• ステッピングモータ (st_pattern)<br>• それ以外(patten) |                                          |                                        |
| 入力    | タクトスイッチ<br>トグルスイッチ                                          | 値の取得、活用                                  |                                        |
|       | PSD 測距センサ                                                   | 値の取得、活用                                  |                                        |
|       | フォトインタラプタ                                                   | パルス値の活用<br>→DC モータ制御                     | パルスカウント<br>パルス数の値の取得                   |
| 出力    | DC モータ                                                      | フラグ管理のみ<br>• 正転 or 逆転<br>• 高速、中速、低速      | カウンタ（1ms 単位のインクリメント）を用いて<br>←フラグに応じて制御 |
|       | ステッピングモータ                                                   | フラグ管理のみ<br>• 正転 or 逆転<br>• 高速、中速、低速      | カウンタ（1ms 単位のインクリメント）を用いて<br>←フラグに応じて制御 |
|       |                                                             | 回転数（ステップ数）の活用<br>→ステッピングモータ制御            | 回転数（ステップ数）のカウント<br>回転数（ステップ数）の取得       |
|       | 圧電ブザー                                                       | フラグ管理のみ<br>• 高音、中音、低音<br>• ピッピ、ピー（音の間隔等） | カウンタ（1ms 単位のインクリメント）を用いて<br>←フラグに応じて制御 |
|       | 7セグメント LED                                                  | 表示データ（左右）の更新<br>表示データ（左右別）の更新            | ←の値に応じて制御（表示）<br>※ダイナミック点灯             |
|       | フルカラー LED                                                   | 表示データ 3bit (RGB) の更新                     | ←の値に応じて制御（表示）                          |

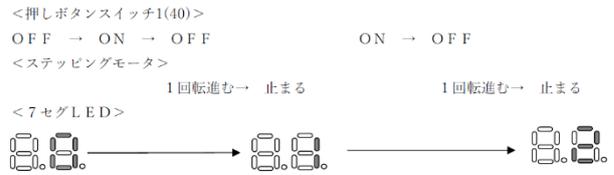
## <サンプルプログラム>

### 動作

ステッピングモータの制御と7セグメントLEDの表示  
押しボタンスイッチ1(40)したとき、ステッピングモータが時計回りに1回転進ようにしなさい。  
また、その時の指針の位置を7セグメントLEDに表示させるようにしなさい。

#### <条件>

- 電源投入時7セグメントLEDは、右の桁のみ[0]が表示されていることとする。
- ステッピングモータの動作開始は、押しボタンスイッチ1(40) OFF時とする。
- 回転中は、トグルスイッチ等の切替の入力を受け付けないこととする。
- 回転数の表示の切り替わりは、回転終了後とする。
- 回転数の表示は、9を最大とする。9の表示のときは、これ以上回転しない。
- 回転方向は、CW方向とする。



## グローバル変数

```
//DCモータ制御用
volatile int8_t dcMotorFlag = 0; //0=停止 1=正転 -1=逆転

//ステップングモータ制御用
volatile int8_t stepTotalPulseCount = 0; //ステップングモータの合計パルス数 1パルスで3度 120パルスで一周
volatile int8_t stepFlag = 0; //0=停止 1=正転 -1=逆転
volatile uint8_t stepSpeed = 10; //ステップングモータのスピード 10以下にすると脱調によりうまく動かなくなる

//圧電ブザー制御用
volatile uint8_t bzTone = 0; //ブザーの音の高さ 0に近づくほど高く
volatile bool bzFlag = false; //trueで鳴る

//7セグメントLED制御用
volatile uint8_t seg10Data, seg1Data; //7セグメント表示データ

//フルカラーLED制御用 (例) 0b100 = 赤色LED点灯 0b010 = 緑色LED点灯 0b001 = 青色LED点灯
volatile uint8_t ledRGB = 0b000; //二進数3ビットでフルカラーLEDを光らせる

//チャタリング対策用 タイマ (カウンタ) (1ms内の割り込みでインクリメント)
volatile uint16_t tyatimer = 0;
```

## メインプログラム( loop()関数内 )

```
static int pattern = 100; //工程管理用のパターン
seg10Data = 0; //7セグメントLEDの左桁のデータ
seg1Data = disp7Seg[abs(stepTotalPulseCount / 120 % 10)]; //7セグメントLEDの右桁のデータ

stepSpeed = 10; //ステップングモータの速度設定(早い)
deltaPulseCount = stepTotalPulseCount - defaultPulseCount; //目標のパルス数までの差

switch (pattern) {
case 100: //ステップングモータ停止
stepFlag = 0; //ステップングモータ停止
if (tyatimer >= 100 && digitalRead(SW40) == LOW) { //一定時間経過後(SWのチャタリング対策)+SW=ON
pattern = 200; //工程(パターン)移行
tyatimer = 0; //チャタリング防止用のタイマカウンタクリア
}
break;

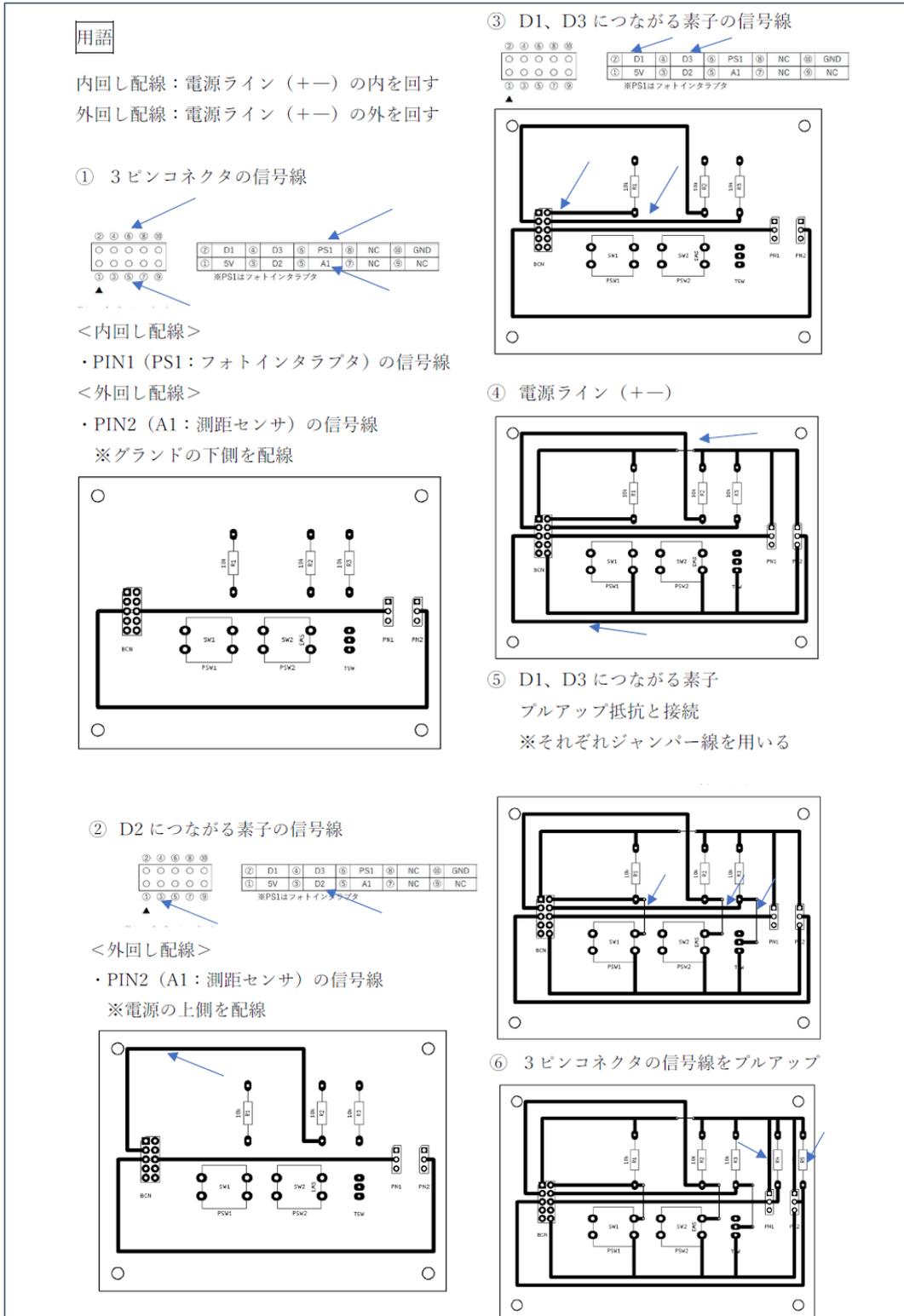
case 200: //スイッチのOFF待ち
if (tyatimer >= 100 && digitalRead(SW40) == HIGH) { //一定時間経過後(SWのチャタリング対策)+SW=OFF
pattern = 300; //工程(パターン)移行
if (stepTotalPulseCount / 120 >= 9) { //9回転以上の時は、回転しない
pattern = 100; //工程(パターン)移行
}
tyatimer = 0; //チャタリング防止用のタイマカウンタクリア
defaultPulseCount = stepTotalPulseCount; //現在のステップングモータの総パルス数を代入
}
break;

case 300:
stepFlag = 1; //ステップングモータ正転
if (abs(deltaPulseCount) >= 120) { //120パルス(1回転)
pattern = 100; //工程(パターン)移行
tyatimer = 0; //チャタリング防止用のタイマカウンタクリア
}
break;
}
```

### 3. 3 回路設計・製作技術の習得

プログラミング同様に、1・2年生にも取り組みやすい方法を考えました。これまでは、各素子の動作や仕組みを学習し、最適（一番早く製作できる）な部品配置やパターン配線を考える練習を行ってきましたが、習得するのに多くの練習時間が必要になると思いました。また、ここで誤った回路を製作してしまうと、『配線組立技術』で減点されるだけでなく、プログラミングを行った際に、仕様通りの動作ができなくなることから、絶対に正しく動く回路を製作できることが重要だと考えました。

そこで、ある程度多くパターン配線を引くことになっても、簡単に確実に製作できるような方法を考え、マニュアル化することを考案しました。また、これを私（3年塩澤）が実際に指導することになりました。※各素子の動作や仕組み、回路チェックの方法については、回路製作が慣れてきたところで、少しずつ時間をかけて説明することになりました。



作成したマニュアル

#### 4. 大会結果（県大会）

県大会の結果は、私（塩澤）が優勝、今年から練習を始めた竹島（2年生）が入賞し、私は長野県代表として北信越大会へ出場することができました。

| 順位 | 氏名    | プログラミング技術<br>(課題点 40点) |             | 配線組み立て技術<br>(課題点 50点) |              | その他<br>(10点) | 合計点  |
|----|-------|------------------------|-------------|-----------------------|--------------|--------------|------|
|    |       | プレ審査<br>(35点)          | 本審査<br>(5点) | 部品実装<br>(30点)         | 設計力<br>(30点) |              |      |
| 1  | 塩澤 匠生 | 16                     | 5           | 26.8                  | 20           | 9            | 76.8 |
| 2  | 松工生徒  | 9                      | 3           | 29.8                  | 20           | 10           | 71.8 |
| 3  | 松工生徒  | 8.5                    | 2           | 29.5                  | 20           | 10           | 70   |
| 4  | 佐久平生徒 | 4                      | 1.6         | 24.7                  | 20           | 9            | 59.3 |
| 5  | 竹島 光星 | 7.5                    | 3           | 25.2                  | 12           | 10           | 57.7 |



競技の様子



表彰式

#### 5. 大会結果（北信越大会）

北信越大会の結果は、私（塩澤）が優秀賞（2位）入賞を果たしました。最後の課題まで私を含め2名がたどり着きましたが、プログラムの完成度で負けてしまいました。

| 順位 | 氏名     | プログラム動作<br>(40点) | 組立技術<br>(30点) | 設計力<br>(20点) | その他<br>(10点) | 合計 |
|----|--------|------------------|---------------|--------------|--------------|----|
| 1  | 砺波工業生徒 | 38               | 29            | 20           | 10           | 97 |
| 2  | 塩澤 匠生  | 34               | 28            | 20           | 10           | 92 |
| 3  | 砺波工業生徒 | 31               | 27.5          | 20           | 10           | 89 |



写真：県大会の結果（表彰）

#### 5. 反省・感想

<塩澤 匠生>

私は今年、ものづくりコンテストの電子回路組み立て部門に全国大会出場を目標に取り組んできました。去年は北信越大会の三位という成績だったので、今年こそ優勝して全国大会に出場できるよう取り組んできました。

去年までの反省を活かし、今年は回路制作技術の向上に力を入れて練習してきました。ジャンパー線をきれいに配線し、半田の量を一定にする練習をたくさんしました。

プログラミングについても今までの経験を活かしさらに正確な動作を行うことを意識し、丁寧なコードを描く練習をしてきました。その結果県大会では優勝できましたが北信越大会では惜しくも二位という結果で終わってしまいました。

全国大会に出場することはできなかったけれど北信越大会当日は自分の中で最高のパフォーマンスを発揮し全力で降り組むことができ、悔いの残らない結果を残せてよかったです。この三年間でプログラミング、はんだ付けの技術共に成長できてよかったです。

#### <竹島 光星>

私はものづくりコンテストを通じて電気回路と制御プログラムの二つの分野について学ぶことができました。電気回路分野については、オームの法則をはじめに、プルアップ回路や抵抗のカラーコードなどの電子部品についても幅広く学ぶことができました。また、はんだ付けの技術も高めることができました。ジャンパー線の配線に関しては、特にきれいに少ない半田でつけていくことを練習してきたため、普通の電子基板へのはんだ付けを以前より簡単に行うことができるようになりました。

また、プログラム分野では電気回路分野に比べてもとても成長できたと感じています。if 文や for 文など、モノづくりコンテスト出場前は全くわからないことだらけでしたが、今では楽々と扱えるまで成長できました。

#### <有賀 啓真>

ものづくりコンテストの電子回路組み立て部門では設計回路の設計、製作、制御プログラムの作成の2つの技術を上達させることができます。

私はその2つの技術を学んだことで次のような技術を得ることができました。

まず、はんだ付け技術です。回路の製作をするにあたってはんだ付け技術は速さ、正確さの2つは必須です。そのため、はんだ付けの技術は上達していきます。

次にプログラム技術です。完成した回路とともに持ってきていたマイコン付きの機体を繋げて指定された動作をできるようにプログラムをします。仕様書を読み、どのような構成にすればその通りに動くかを考えるため、どのように書けば自分の想像する通りに動くかという感覚が身に着きます。

ものづくりコンテストで身に付けた技術を生かして、マイコンカーなどの機体の製作や動作させるためのプログラムを書くことなどに生かしていきたいと思います。

#### <花岡 良太郎>

今回、私は二回目の参加となったが前回としばらく間が空いていたため練習のうち、最初は去年どんな感じでやってたかを考えながら練習していました。そして、しばらく練習を重ねていくうちに「ここはこんな考え方だったな」や「こうやった方がやりやすかったな」といった風に去年の技術を思い出して応用することができました。また、プログラムにおいては去年、モノコンでプログラムの基礎的な部分を学びそこから日々の部活動でさらに上達し今年のモノコンで更に磨きをかけることで上達しました。

大会の練習で磨いた技術を日々の部活動でのプログラムや授業で友達に教えるときなど様々な場面に活かすことができるはずなので活かしていきたいと思います。

#### <小松 眞丈>

ものづくりコンテストの電子回路部門に参加し、さまざまな経験をしました。

まず、回路設計の基礎から実装までの一連の流れを深く理解することができました。回路図の作成やはんだ付け、部品の配置など、実践的な技術を身につけることができ、大変勉強になりました。実際に回路を組み立てる中で、設計通りに動作しないことも多くありましたが、その原因を特定し、修正を重ねることで技術を向上させることができました。

また、プログラミングを学ぶことで、コードを書く際の処理の流れを考える習慣が身につき、論理的思考力を養うことができました。特に、条件分岐やループ処理、関数の活用を理解することで、より効率的にコードを書く力が身につきました。

これらの経験を通じて得た知識や技術は、今後の学習や実践に役立つと考えています。さらに技術と理解を深めていきたいです。

#### <東 馳斗>

私がものづくりコンテストで習得した技術は、回路のはんだ付け技術と、プログラミングの技術です。その中でも、今の私に一番影響があるのははんだ付けの技術です。

私は入部当初、はんだ付けがあまり得意ではなく、抵抗等の実装でも位置が曲がってしまったり、ピンソケットの実装では、高さに偏りが出たりしてしまう事もありました。しかし、練習を始めてから、私は数多くの経験を積み、今では綺麗に適切なはんだ量で実装出来るようになってきました。実際、大会成績は下位から数えて2番目という結果でしたが、回路点数のみであれば、6位の成績に並ぶ実力を得ることができました。

この技術は、大会に挑戦し、得たものの中で最も大きな物だと思っています。その他にも、私は多くの技術を習得する事ができました。

来年度は、5位以内を目指して活動を続けて行きます。