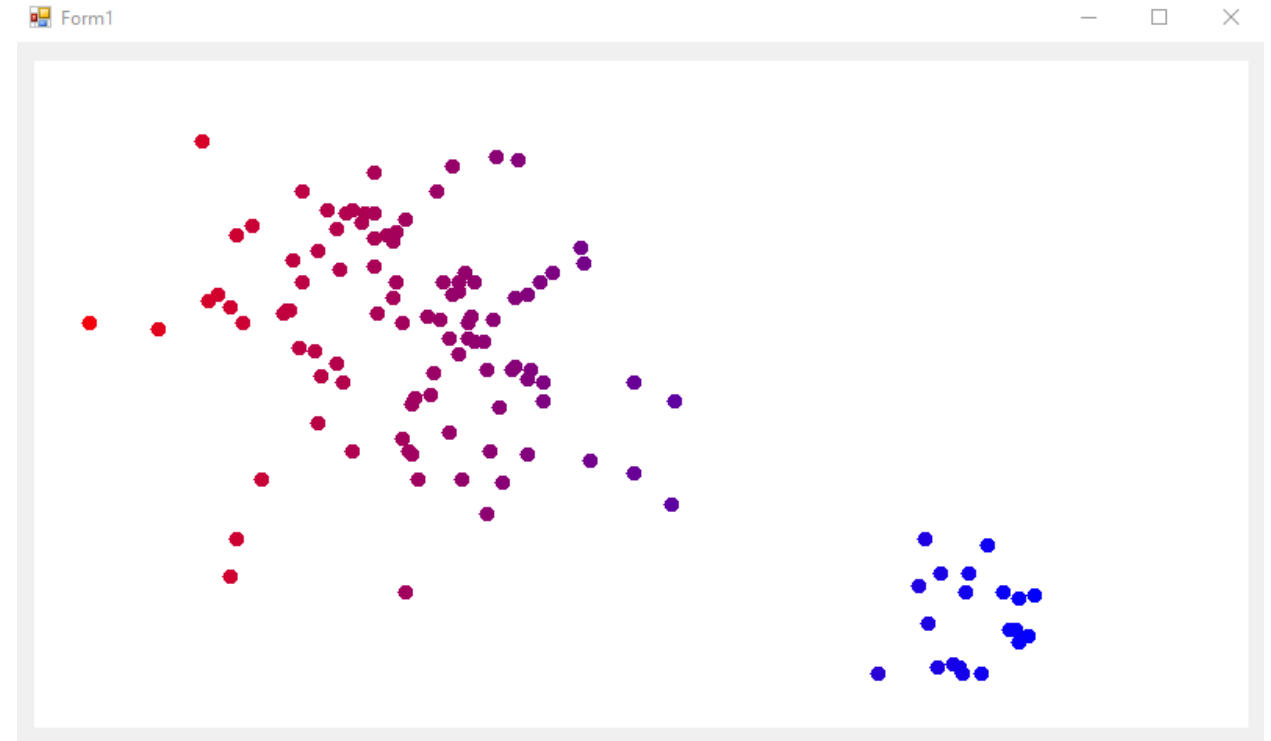


9回目(6月28日)

- アプリケーション作成

8回目

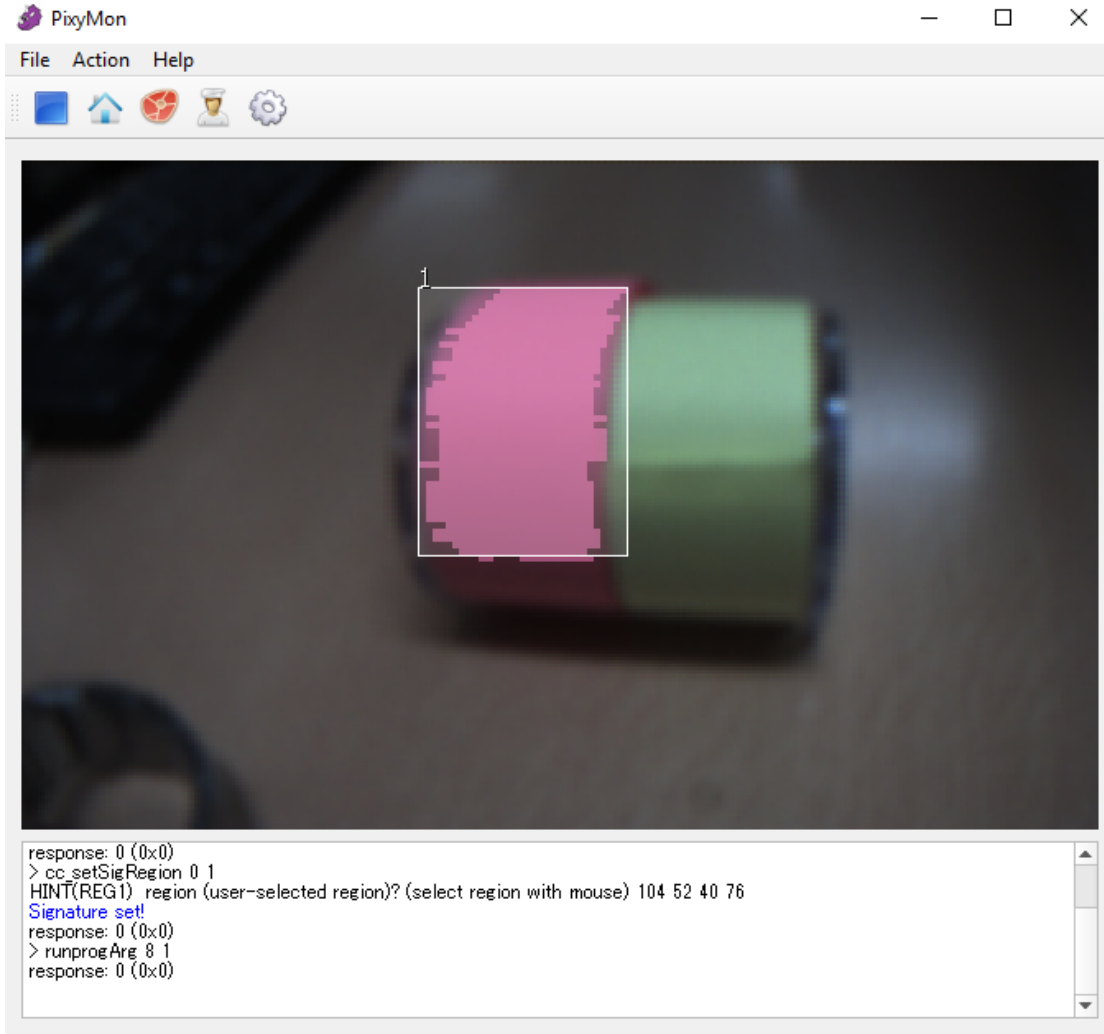
- Sampleアプリケーション作成



- 設定した色をUIで出ているというアプリ

ステップ 1

- PixyMonでSet Signatureする



- ちゃんと一つ色だけで設定する
- Backgroundと違う色がおすすめ

ステップ2

- Arduinoでプログラムする
 - PBL2018-students/Arduino-sample/for_application_03に見てください
- Loop()の中にそれが入ってる

```
static int i = 0;
int j;
uint16_t blocks;
char buf[100];

// grab blocks!
blocks = pixy.getBlocks();

// If there are detect blocks, print them!
if (blocks)
{
    i++;

    // do this (print) every 50 frames because printing every
    // frame would bog down the Arduino
    if (i%20==0)
```

```
{
    sprintf(buf, "%d ", blocks);
    Serial.print(buf);
    for (j=0; j<blocks; j++)
    {
        if(pixy.blocks[j].angle)
        {
            sprintf(buf, "c %d ", pixy.blocks[j].angle);
        }
        else
        {
            sprintf(buf, "s ");
        }
        Serial.print(buf);

        sprintf(buf, "%d %d %d %d %d ", pixy.blocks[j].signature, pixy.blocks[j].x, pixy.blocks[j].y,
pixy.blocks[j].width, pixy.blocks[j].height);
        Serial.print(buf);
        sprintf(buf, "");
    }
    Serial.print("end¥n");
}
}
```

ステップ3

- ArduinoでVerifyとuploadする
- Serial Monitorで確認

COM3 (Arduino Leonardo)

```
3 s 1 238 83 101 31 s 1 203 109 23 3 s 1 250 112 21 2 end
```

これ見たい感じ出てたらOK!

Autoscroll

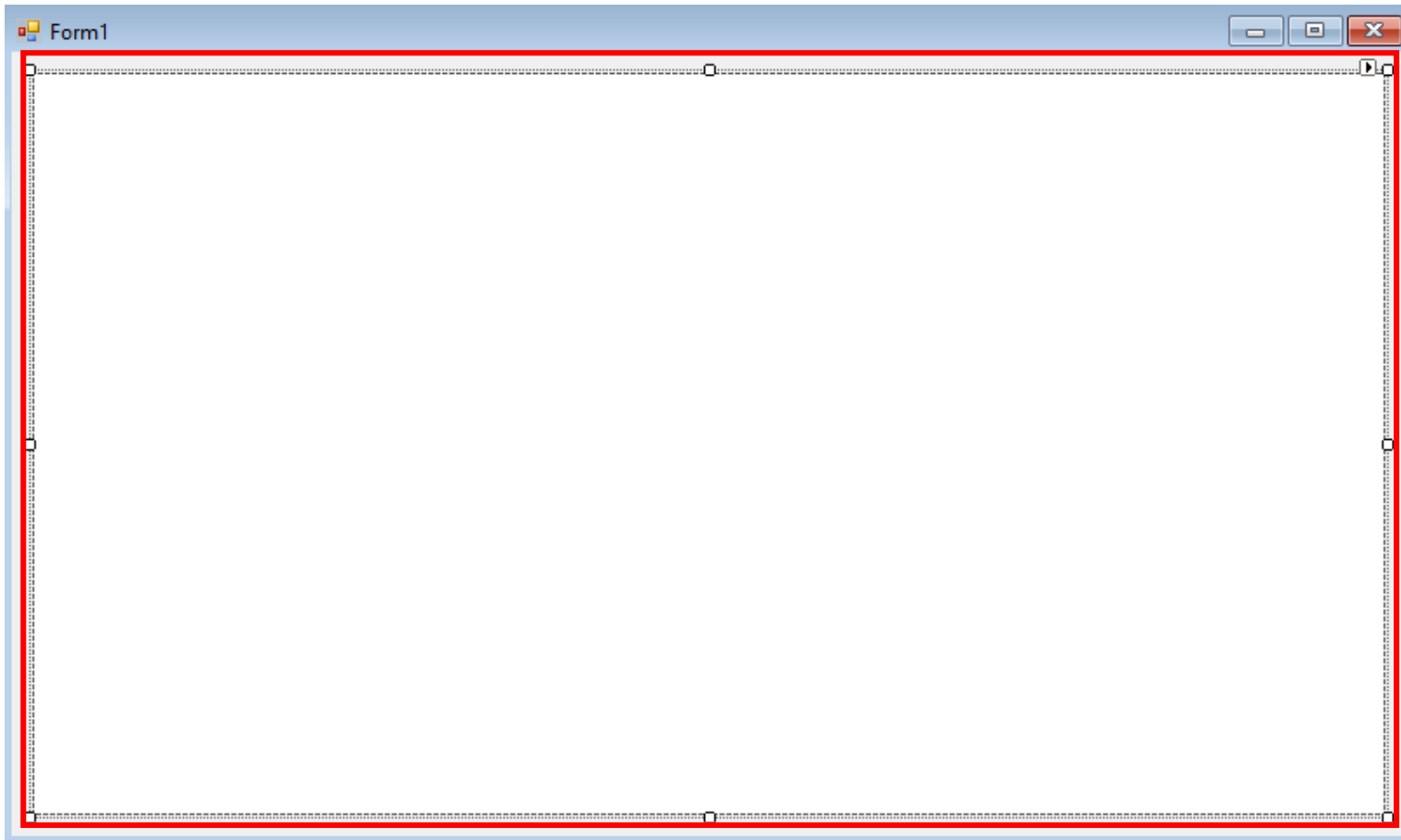
No line ending ▾

9600 baud ▾

Clear output

ステップ4

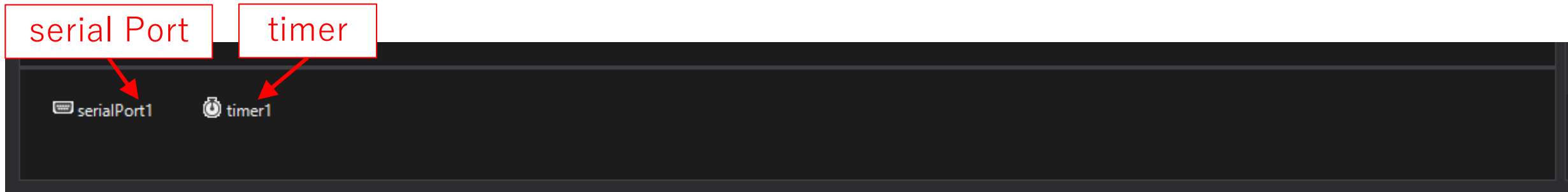
- Visual Studioでプログラム
- まず新しいプロジェクト作って、Form作成



PictureBox

ステップ5/1

- タイマとSerial Port入れて



- タイマクリック

```
private void timer1_Tick(object sender, EventArgs e)
{
    String s;
    String s_temp;
    int byte_to_read = 0;
    int d_num;
    int at = 0;
    int i, j;
    int[] ob = new int[6]; // (0:角度 (単色のときは500)、1:Signatuer番号、2:x座標、3:y座標、4:横幅、5:高さ)

    if (serialPort1.IsOpen == true)
    {
        byte_to_read = serialPort1.BytesToRead;
    }
}
```


ステップ5/2

```
//シリアル通信で文字列が送られてきたらif文を実施
if (byte_to_read > 5)
{
    //Arduinoからシリアル通信で送られてくる文字列をsに格納する
    s = serialPort1.ReadTo("end\r\n");

    //検出された領域の個数をd_numに格納する
    at = 0;
    while (at == 0 || at == -1)
    {
        at = s.IndexOf(' ');
    }
    s_temp = s.Remove(at);
    s = s.Remove(0, at + 1);
    Int32.TryParse(s_temp, out d_num);

    //検出された個数分繰り返す
    for (i = 0; i < d_num; i++)
    {
        //先頭の文字を格納
        at = 0;
        while (at == 0 || at == -1)
        {
            at = s.IndexOf(' ');
        }
        s_temp = s.Remove(at);
        s = s.Remove(0, at + 1);
    }
}
```

ステップ5/3

```
//先頭が「c」だったらb[0]に角度 (-180~180)を格納。「s」だったら、b[0]500を格納
if ((String.Compare(s_temp, "c") == 0) || (String.Compare(s_temp, "s") == 0))
{
    if (String.Compare(s_temp, "c") == 0)
    {
        at = 0;
        while (at == 0)
        {
            at = s.IndexOf(' ');
        }
        s_temp = s.Remove(at);
        Int32.TryParse(s_temp, out ob[0]);
        s = s.Remove(0, at + 1);
    }
    else
    {
        ob[0] = 500;
    }

    //ob[1]~ob[5]に、Signature番号、x座標、y座標、横幅、高さを順に格納
    for (j = 1; j <= 5; j++)
    {
        at = 0;
        while (at <= 0)
        {
            at = s.IndexOf(' ');
        }
        s_temp = s.Remove(at);
        s = s.Remove(0, at + 1);
        Int32.TryParse(s_temp, out ob[j]);
    }
}
```

ステップ5/4

```
Graphics gr = pictureBox1.CreateGraphics();

// Pen redPen = new Pen(Color.Red, 5);

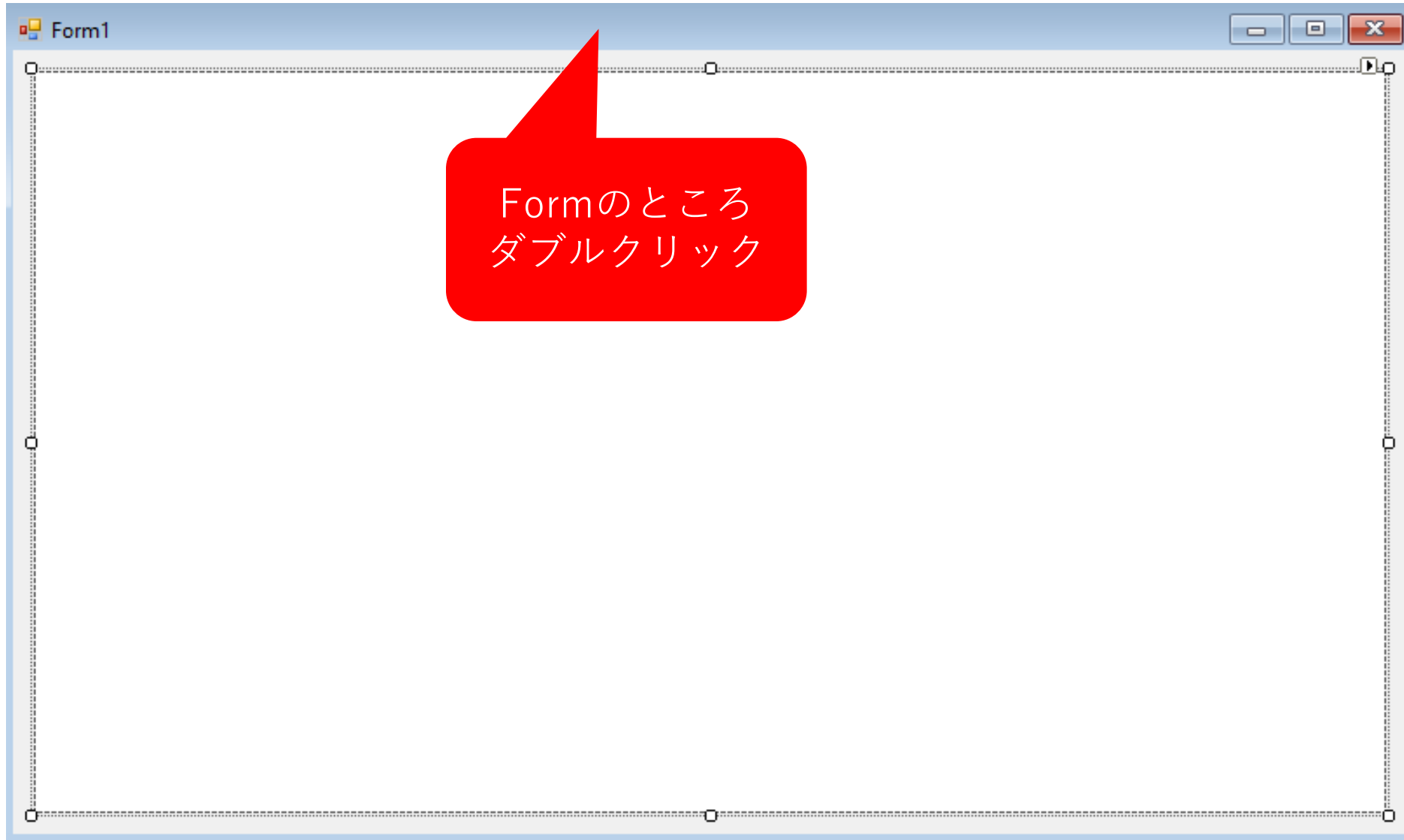
SolidBrush myBrush = new SolidBrush(Color.FromArgb(255 * ob[2] / 320, 0, 255 - 255 * ob[2] / 320));

//ob[2]とob[3]に格納されたxy座標に点を描く
// gr.DrawLine(redPen, 640 - ob[2] * 2, ob[3] * 2, 640 - ob[2] * 2 + 2, ob[3] * 2 + 2);

GraphicsPath graphPath = new GraphicsPath();
graphPath.AddEllipse(640 - ob[2] * 2, ob[3] * 2, 10, 10);
gr.FillPath(myBrush, graphPath);
}
}
}
```

- この上にcodeいれてください (後で説明する)

ステップ6



Formのところ
ダブルクリック

ステップ7

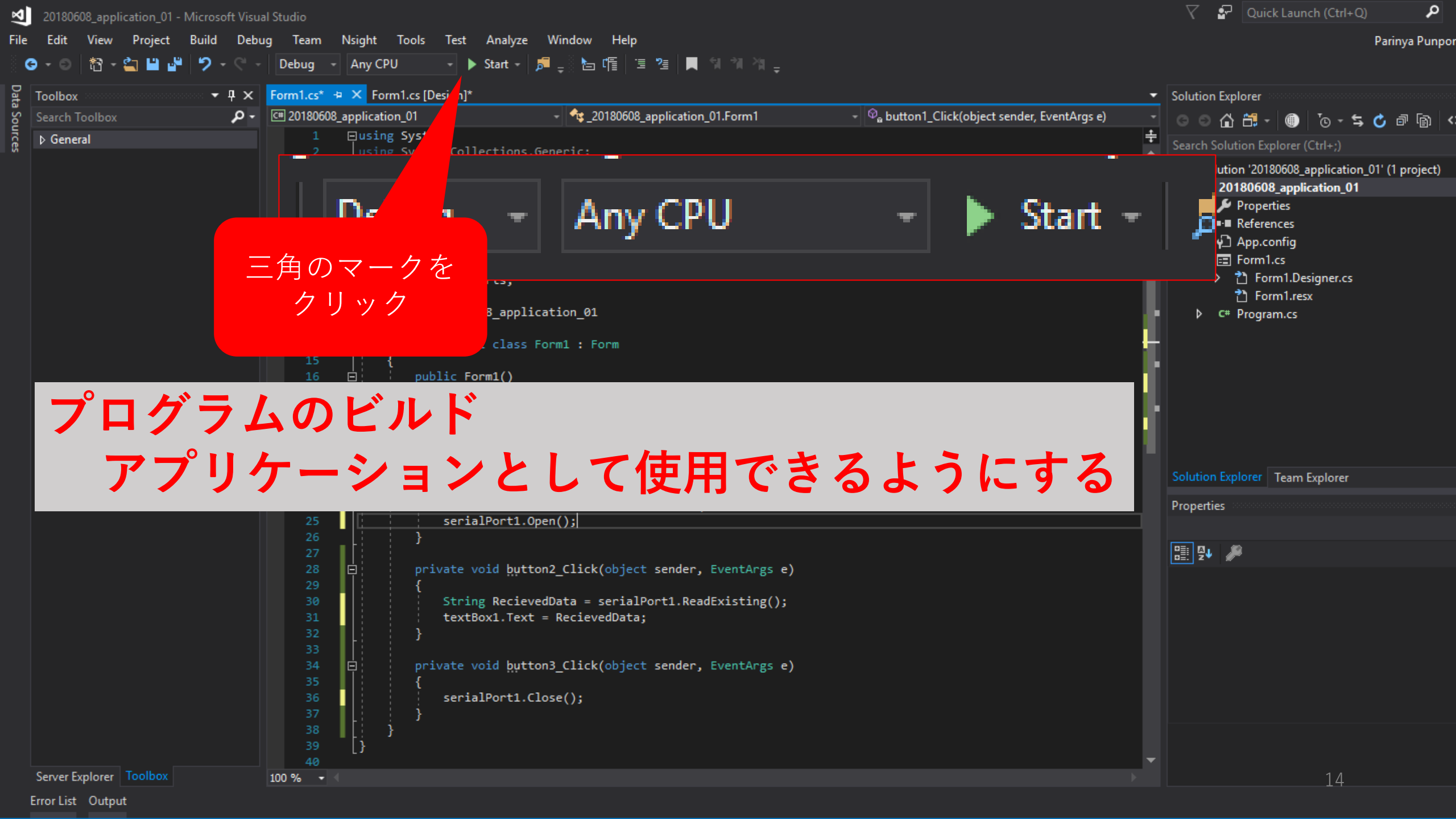
- このcodeいれてください (後で説明する)

```
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Interval = 1;
    timer1.Enabled = true;

    try
    {
        serialPort1.PortName = "COM00";
        serialPort1.Handshake = Handshake.RequestToSend;
        serialPort1.Open();
        serialPort1.ReadTimeout = 2000;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        Application.Exit();
    }
}
```

Arduinoと同じ





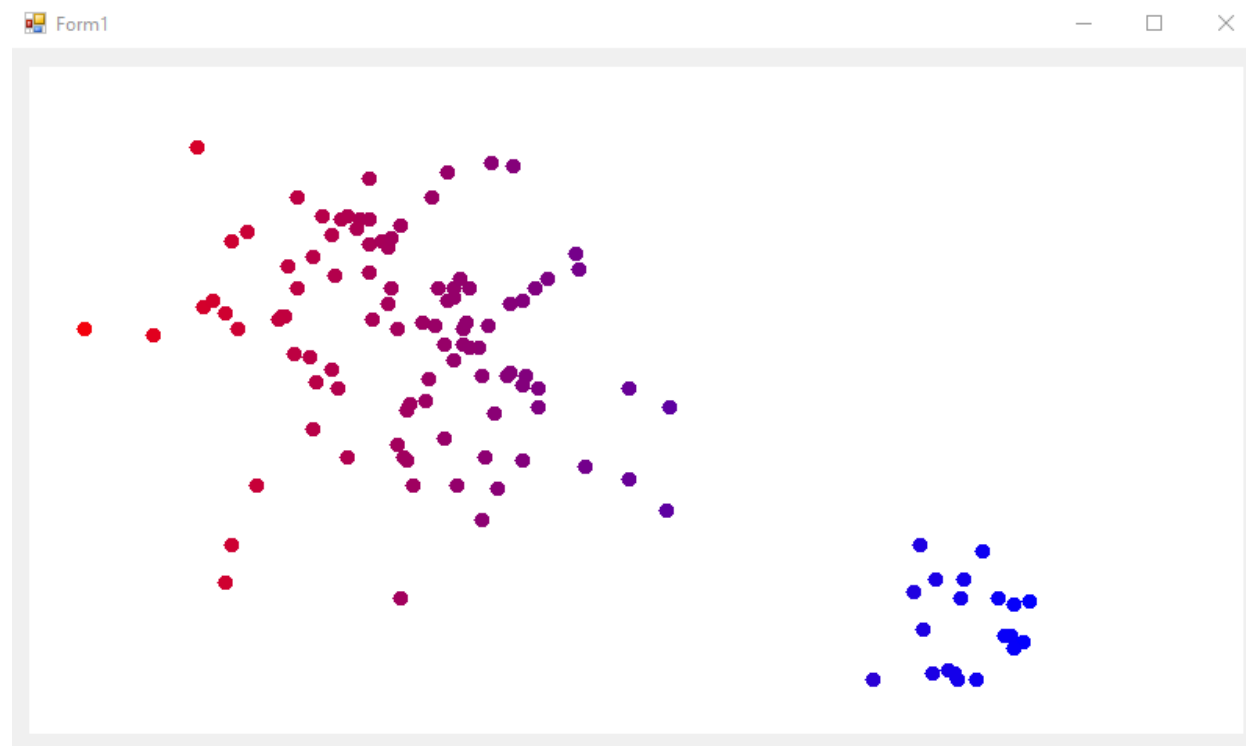
三角のマークを
クリック

プログラムのビルド
アプリケーションとして使用できるようにする

```
25 serialPort1.Open();  
26 }  
27  
28 private void button2_Click(object sender, EventArgs e)  
29 {  
30     String RecievedData = serialPort1.ReadExisting();  
31     textBox1.Text = RecievedData;  
32 }  
33  
34 private void button3_Click(object sender, EventArgs e)  
35 {  
36     serialPort1.Close();  
37 }  
38 }  
39  
40
```

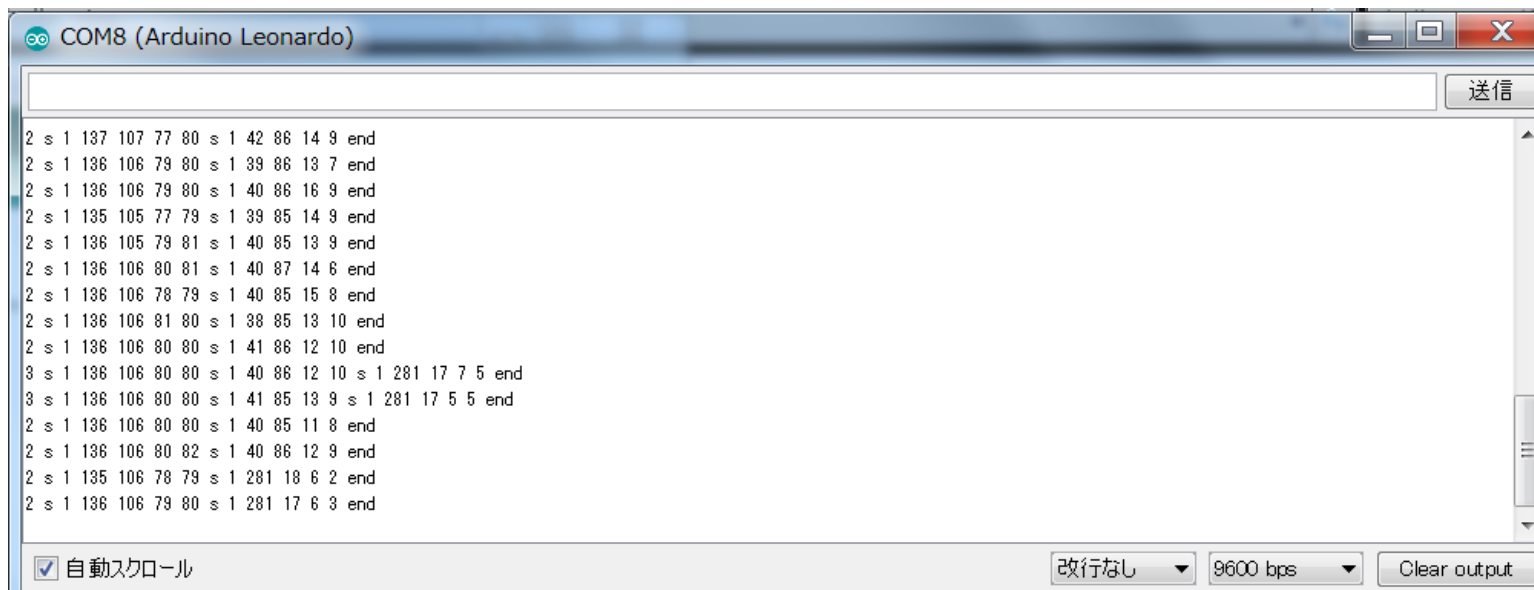
テスト

- 設定したいろをカメラ見つけたら、UIの画像で出ている



できない場合は ステップ2を確認してください

1. Arduinoアップロードしてない？
2. ArduinoでSerial Monitorは何も出てない？



The screenshot shows the Arduino Serial Monitor window titled "COM8 (Arduino Leonardo)". The window contains a list of data points, each consisting of two sets of four numbers separated by a semicolon, followed by "end". The data points are as follows:

```
2 s 1 137 107 77 80 s 1 42 86 14 9 end
2 s 1 136 106 79 80 s 1 39 86 13 7 end
2 s 1 136 106 79 80 s 1 40 86 16 9 end
2 s 1 135 105 77 79 s 1 39 85 14 9 end
2 s 1 136 105 79 81 s 1 40 85 13 9 end
2 s 1 136 106 80 81 s 1 40 87 14 6 end
2 s 1 136 106 78 79 s 1 40 85 15 8 end
2 s 1 136 106 81 80 s 1 38 85 13 10 end
2 s 1 136 106 80 80 s 1 41 86 12 10 end
3 s 1 136 106 80 80 s 1 40 86 12 10 s 1 281 17 7 5 end
3 s 1 136 106 80 80 s 1 41 85 13 9 s 1 281 17 5 5 end
2 s 1 136 106 80 80 s 1 40 85 11 8 end
2 s 1 136 106 80 82 s 1 40 86 12 9 end
2 s 1 135 106 78 79 s 1 281 18 6 2 end
2 s 1 136 106 79 80 s 1 281 17 6 3 end
```

The window also features a "送信" (Send) button at the top right, a "自動スクロール" (Auto scroll) checkbox checked at the bottom left, and a "Clear output" button at the bottom right. The baud rate is set to 9600 bps and the line ending is set to "改行なし" (No line ending).

アプリケーション作成内容レポート

簡単に**A4の1ページ以内**書いてください

A: どんなアプリケーション作成する？

B: 分かりやすい画像/図画

名前

名前

名前

アプリケーション名前

内容内容内容内容内容

内容内容内容内容内容

内容内容内容内容内容

内容内容内容内容内容

図画

6/27 (水) までに、以下のPDFにメールを下さい。

parinya@sys.es.osaka-u.ac.jp

*メールのtitleは「**[PBL2018]-0000000000000000**」

来週から

**レポート出した内容を
グループでアプリケーション作成する** (先生と相談可能)

- 7月5日 アプリケーション作成
- 7月12日 アプリケーション作成 / 写真・動画レクチャー
- 7月19日 アプリケーション作成
(夏休み：Free time to finishアプリケーション)
- 9月13日 佐藤研内プレゼンテーション(TBD)
- 9月27日 知能システム学コース最終発表会**

相談

佐藤研以内誰でも、先生だと2人が担当

- 松倉 悠（助教）
- Parinya Punpongsanon（助教）

相談したいなら一日前にメールしてください！

parinya@sys.es.osaka-u.ac.jp

（松倉先生相談でも私までメール）

来週からの時間？

7月5日 アプリケーション作成

7月12日 アプリケーション作成 / 写真・動画レクチャー

7月19日 アプリケーション作成

(夏休み：Free time to finishアプリケーション)

何時？何曜日？