Design

TOF

Confidential

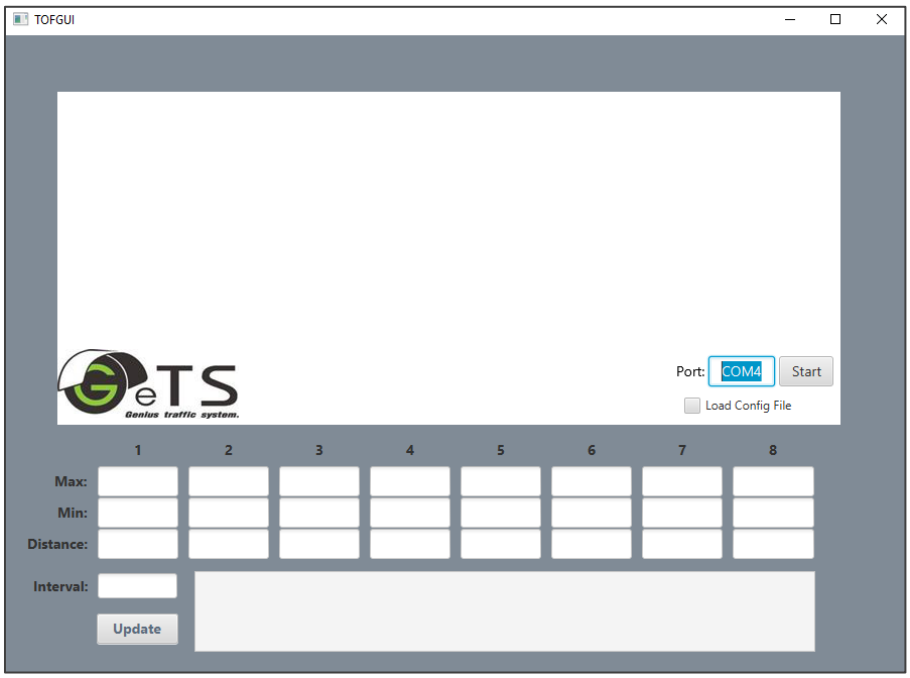
(2016-12-6)

Draft

Requirements

- โปรแกรมสามารถแสดงระยะห่างระหว่าง controller กับคนที่จะข้ามถนน (เซนติเมตร)
- โปรแกรมสามารถปรับค่า Min และ Max ของบอร์ดได้
- สามารถใช้งานได้กับบอร์ดที่มีเซ็นเซอร์ 8 และ 16 เส้น
- User Interface ใช้งานง่าย

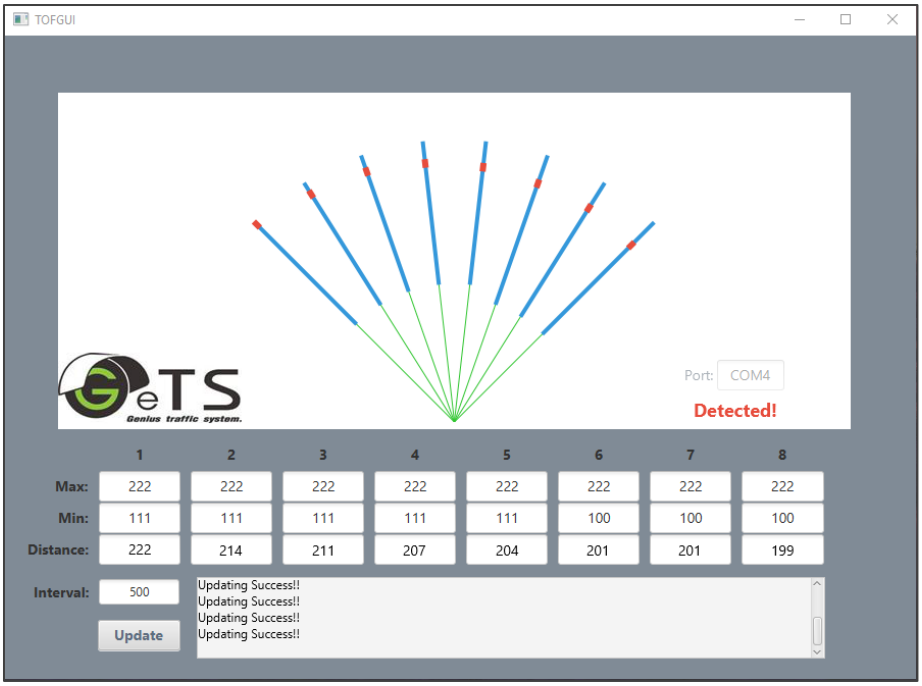
Program Overview



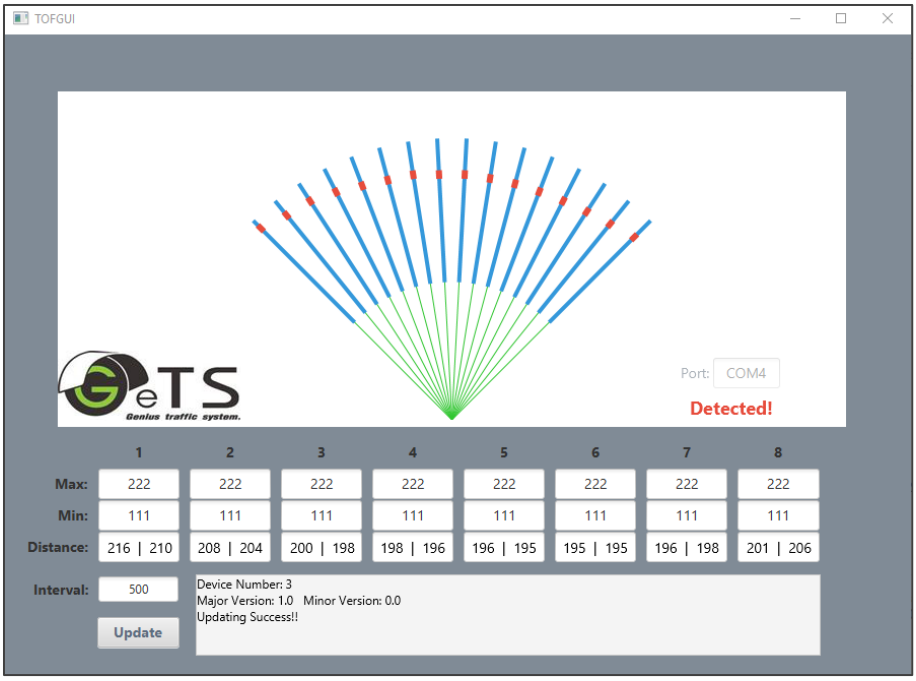
ภาพที่ 1: ภาพแสดงตัวอย่างหน้าต่างการทำงานของโปรแกรมเมื่อผู้ใช้เริ่มเปิดโปรแกรม

จากภาพด้านบนเมื่อผู้ใช้เปิดโปรแกรมก็จะต้องทำการเลือก Port ที่ Arduino ต่ออยู่ นอกจากนั้นผู้ใช้ยังสามารถเลือกได้ว่าจะใช้ค่าMin และ Max เดิมที่บันทึกอยู่ใน Arduino หรือจะโหลดค่า Min และ Max จากไฟล์ config.txt ก็ได้ ซึ่งทำได้โดยคลิกที่กล่อง Checkbox ด้านล่าง

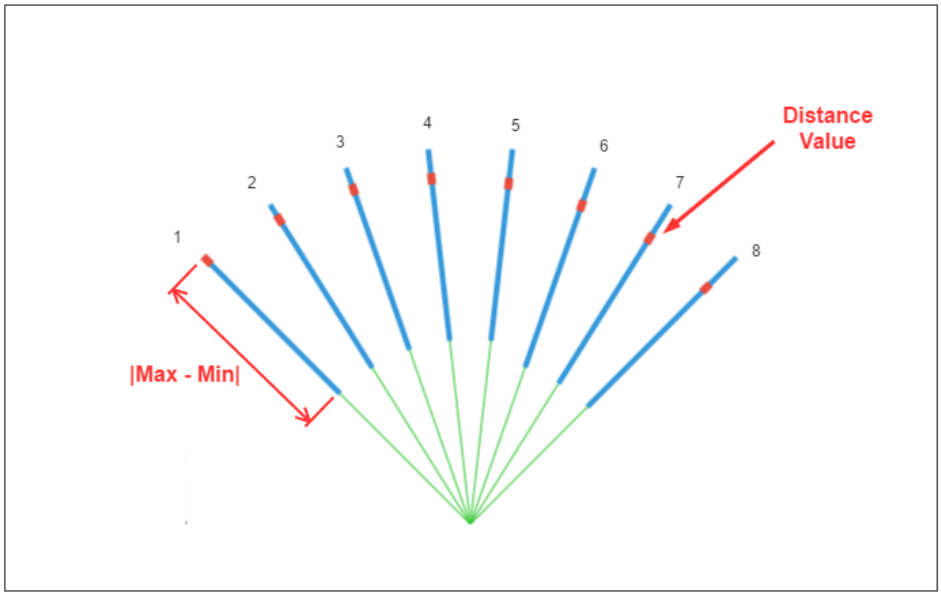
เมื่อเลือก Port เสร็จแล้วให้คลิกที่ปุ่ม Start เพื่อเริ่มการติดต่อกับบอร์ด Arduino ซึ่งตัวโปรแกรมนี้สามารถใช้งานได้กับบอร์ด Arduino ที่มีเซ็นเซอร์ทั้ง 8 และ 16 เส้น ซึ่งหน้าตาโปรแกรมก็จะแตกต่างกันออกไปตามภาพที่2 และ3 ตามลำดับ



ภาพที่ 2: ภาพแสดงตัวอย่างหน้าต่างการทำงานของโปรแกรมเมื่อเชื่อมต่อกับบอร์ด Arduino ที่มี Sensor 8 เส้น

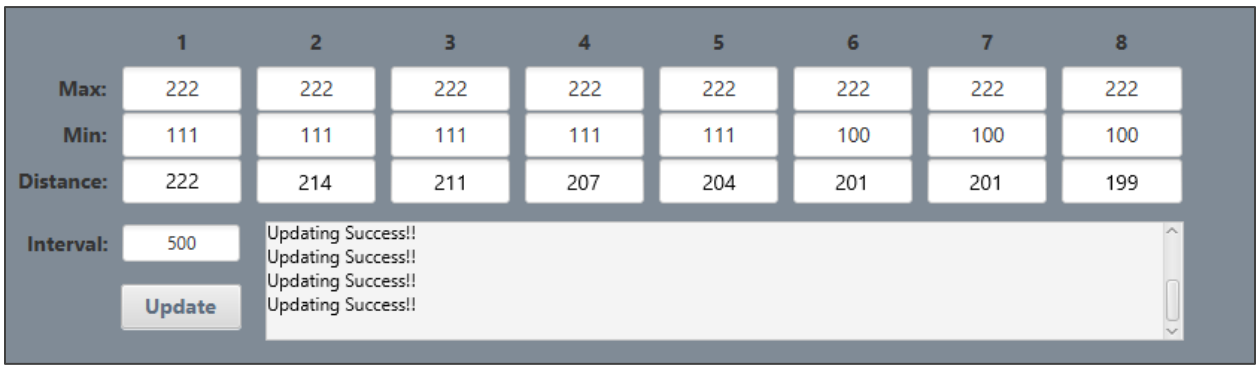


ภาพที่ 3: ภาพแสดงตัวอย่างหน้าต่างการทำงานของโปรแกรมเมื่อเชื่อมต่อกับบอร์ด Arduino ที่มี Sensor 16 เส้น



ภาพที่ 4: ภาพอธิบายความหมายของเส้นบนจอ Monitor

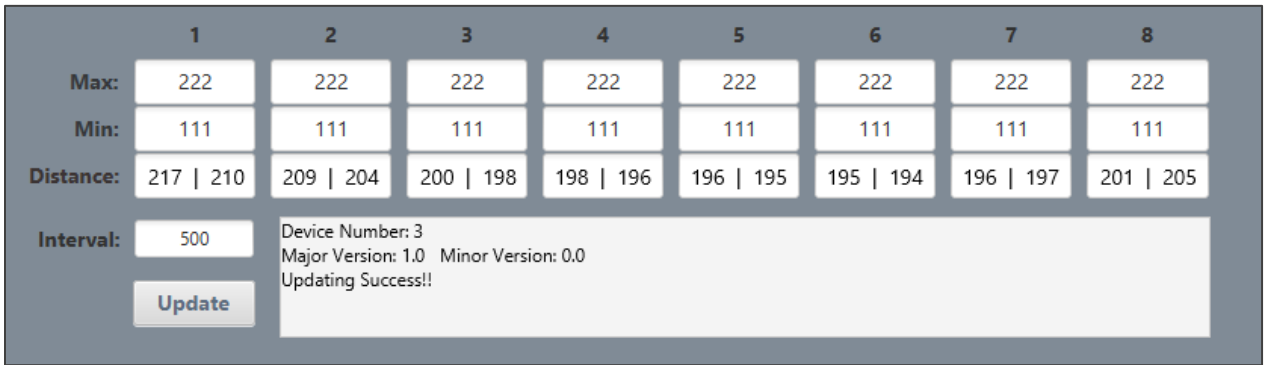
จากภาพด้านบนแสดงระยะทางทั้งหมดที่ Sensor อ่านได้ ซึ่งจุดสีแดงที่เห็น(Distance Value) แสดงถึงค่าระยะทางที่ Sensor แต่ละเส้นนั้นตรวจพบวัตถุ และเส้นสีน้ำเงินแสดงถึงขอบเขตของค่า Min และค่า Max ของบอร์ด Arduino โดยระยะทางจากตำแหน่งปลายสุดของเส้นทุกเส้น(ค่า Max) ถึงจุดล่างสุดจะมีระยะทางคงที่เสมอ



ภาพที่ 5: ภาพแสดงแผงควบคุมในโปรแกรม เมื่อต่อเข้ากับของบอร์ด Arduino ที่มี Sensor ทั้งหมด 8 เส้น

จากภาพด้านบน แถวแรกและแถวที่สอง คือแถวที่แสดงค่า Max และค่า Min ของบอร์ดตามลำดับ ซึ่งผู้ใช้ต้องการจะเปลี่ยนค่า Min และ Max ของบอร์ด ก็สามารถแก้ไขค่าดังกล่าวได้ตามใจชอบ และกดปุ่ม “Update” เพื่อทำการส่งค่า Min และ Max ดังกล่าวไปยังบอร์ดนั้นเอง และเมื่อบอร์ดถูกเซ็ตค่า Min และ Max เรียบร้อยแล้ว โปรแกรมก็จะแสดงสถานะ “Updating Success!!” ให้ทราบ

ส่วนแถวที่ 3 นั้นคือค่า Distance หรือค่าระยะทางที่ Sensor ทุกตัวอ่านได้ (หน่วยเป็นเซนติเมตร) ซึ่งค่า Distance ทั้งหมดนี้ก็จะอัพเดททุกๆช่วงเวลาตามค่า Interval นั้นเอง (ในที่นี้ใช้ค่า Interval = 500 millisecond หรือ ½ วินาทีนั้นเอง)



ภาพที่ 6: ภาพแสดงแผงควบคุมในโปรแกรม เมื่อต่อเข้ากับของบอร์ด Arduino ที่มี Sensor ทั้งหมด 16 เส้น

จากภาพด้านบนแสดงให้เห็นถึงแผงควบคุมบอร์ด Arduino แบบ 16เส้น ซึ่งจะมีความแตกต่างจาก แผงที่ใช้ควบคุม Arduino 8 เส้น อยู่เล็กน้อย

อันดับแรกคือ แถว Distance จะมีตัวเลขทั้งหมด 16 ตัว ซึ่งแทนค่าระยะทางของเซ็นเซอร์ทั้ง 16 ตัว เรียงลำดับจากซ้ายไปขวา

อย่างที่สองคือ ในการเซ็ตค่า Min และ Max นั้นเมื่อผู้ใช้เซ็ตค่าที่ช่อง 1 ค่าMin และ Max ของบอร์ดที่จะถูกเปลี่ยนคือ ค่าMin และ Max ของเซ็นเซอร์เส้นที่ 1 และ 2 และถ้าผู้ใช้เซ็ตค่าที่ช่อง 2 ค่าMin และ Max ของบอร์ดที่จะถูกเปลี่ยนคือ ค่าMin และ Max ของเซ็นเซอร์เส้นที่ 3 และ 4 แบบนี้ไปเรื่อยๆนั้นเอง

Protocol

การเชื่อมต่อ CPU โดยอุปกรณ์ที่ทำการเชื่อมต่อผ่านทาง RS232 โดยตั้งค่าดังต่อไปนี้

Bits per second : 115200
Data bits : 8
Parity : None
Stop bits : 1
Flow control : None

** ในการส่งข้อมูลไปแต่ละไบต์ไปตู้ Controller ให้ delay ระหว่างไบต์อย่างน้อย 60 millisecond*

Computer ทำการส่งคำสั่งเพื่อร้องขอไปยัง CPU แล้ว CPU จึงตอบกลับมา

Commands (16 เส้น)

รหัส	คำสั่ง	รายละเอียด
0	Get Version	ขอข้อมูลหมายเลขเครื่องและ version
1	Send Data	ส่ง Min[16] และ Max[16] ที่อ่านได้จาก config.txt ไปยัง Arduino
2	Get Data	รับ Min[16] และ Max[16] จาก Arduino
3	Get Distance	รับค่า Distance[16] ที่ได้จากเซ็นเซอร์

1. Get version

ส่งคำสั่งไปยัง Arduino เพื่อขอรับข้อมูลข้อมูลหมายเลขเครื่องและ version จาก Arduino

Computer -> Arduino (4 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	1	รหัสคำสั่ง = 0
2	CRC8	1	CRC8
3	End	1	'}'

Arduino -> Computer (7 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	1	รหัสคำสั่ง = 0
2	Device number	1	หมายเลขเครื่อง เช่น 3 หมายถึง เซ็นเซอร์ 16 เส้น 4 หมายถึง เซ็นเซอร์ 8 เส้น
3	Major version	1	เลขตัวหน้า เช่น version 1.0 , major version คือ 1
4	Minor version	1	เลขตัวหลัง เช่น version 1.0 , minor version คือ 0
3	CRC8	1	CRC8
4	End	1	'}'

2. Send Data

ส่งค่า Min และ Max ไปยัง Arduino เพื่อเซ็ตค่า Min และ Max ของ Arduino ใหม่

Computer -> Arduino (68 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	1	รหัสคำสั่ง = 1

2-33	Min	32	ค่า Min[16] ที่ส่งไปให้บอร์ด
			min[HI] = (byte)((min >> 8) & 0xFF);
			min[Lo] = (byte)(min & 0xFF);

34-65	Max	32	ค่า Max[16] ที่ส่งไปให้บอร์ด
			max[HI] = (byte)((max >> 8) & 0xFF);
			max[Lo] = (byte)(max & 0xFF);

66	CRC16	1	CRC8
67	End	1	'}'

ภาพที่ 7: ภาพแสดงโครงสร้างคำสั่ง Send Data

Arduino -> Computer (5 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	1	รหัสคำสั่ง = 1
2	Status	1	สถานะปกติ = 0
3	CRC8	1	CRC8
4	End	1	'}'

3. Get Data

ส่งคำสั่งไปยัง Arduino เพื่อขอรับค่า Min[16] และค่า Max[16] ปัจจุบันของ Arduino

Computer -> Arduino (4 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	2	รหัสคำสั่ง = 2
2	CRC8	1	CRC8
3	End	1	'}'

Arduino -> Computer (68 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	2	รหัสคำสั่ง = 2
2-33	Min	32	ค่า Min[16] ที่อ่านจาก EEPROM ตำแหน่งที่ 0-31
34-65	Max	32	ค่า Max[16] ที่อ่านจาก EEPROM ตำแหน่งที่ 32-63
66	CRC16	1	CRC8
67	End	1	'}'

ภาพที่ 8: ภาพแสดงโครงสร้างคำสั่ง Get Data

4. Get Distance

ส่งคำสั่งไปยัง Arduino เพื่อขอรับค่า Distance[16] ที่ Arduino ได้รับจาก Sensors ทั้งหมด

Computer -> Arduino (4 bytes)

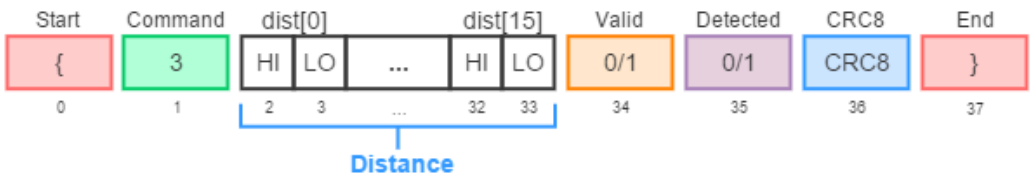
ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	3	รหัสคำสั่ง = 3
2	CRC8	1	CRC8
3	End	1	'}'

Arduino -> Computer (38 bytes)

ลำดับ	ชื่อ	ความยาว (byte)	รายละเอียด
0	Start	1	‘{’
1	Command ID	1	รหัสคำสั่ง = 3

2-33	Distance	32	ค่า Distance[16] ที่ได้จากเซ็นเซอร์
			b[HI] = (byte)((c >> 8) & 0xFF); // CRC16 [HI]
			b[Lo] = (byte)(c & 0xFF); // CRC16 [LO]

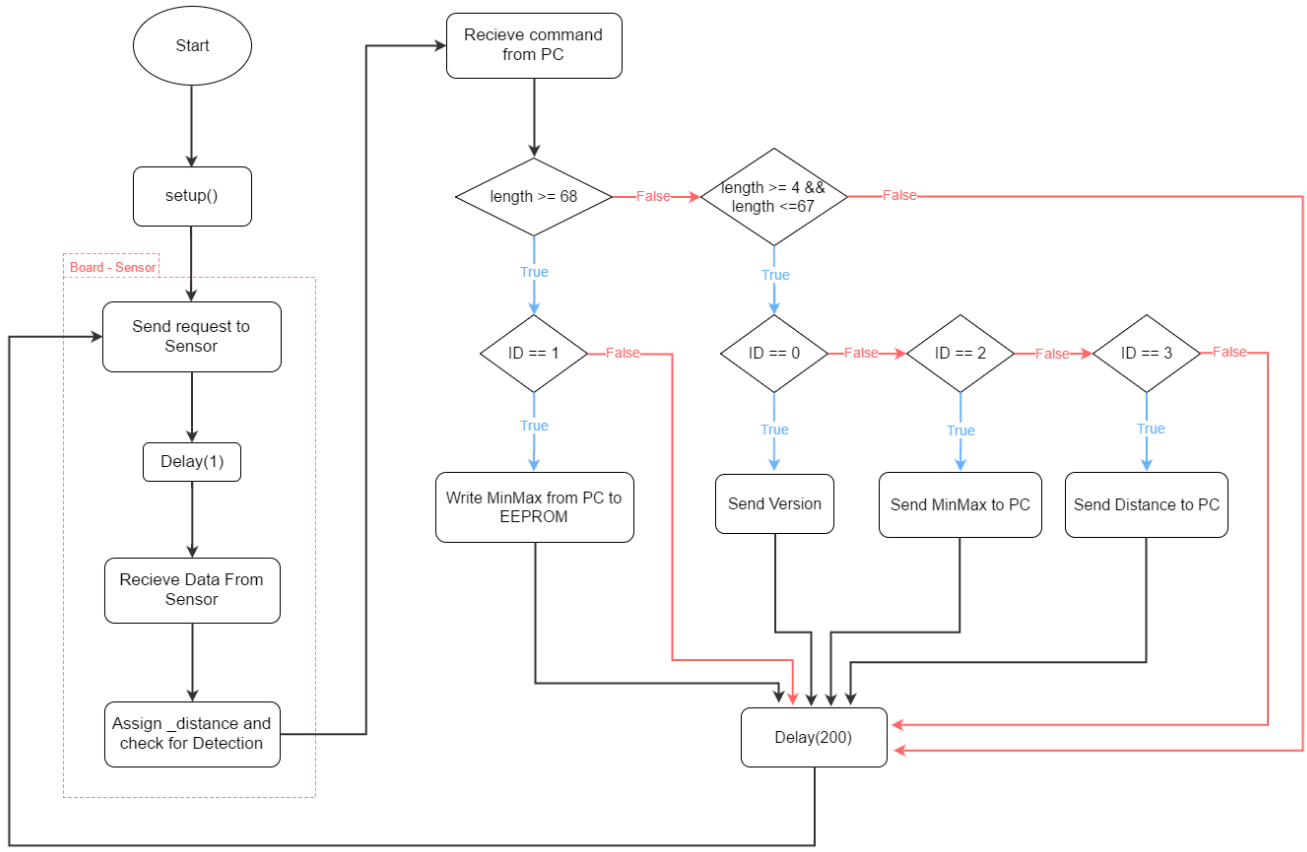
34	valid	1	ได้รับข้อมูลจาก sensor 0 = ไม่ได้รับข้อมูล 1 = ได้รับข้อมูล
35	detected	1	ตรวจพบวัตถุ 0 = ไม่พบวัตถุ 1 = ตรวจพบวัตถุ
36	CRC8	1	CRC8
37	End	1	'}'



ภาพที่ 9: ภาพแสดงโครงสร้างคำสั่ง Get Distance

Flow Chart (16 เส้น)

1. ฝั่ง Arduino



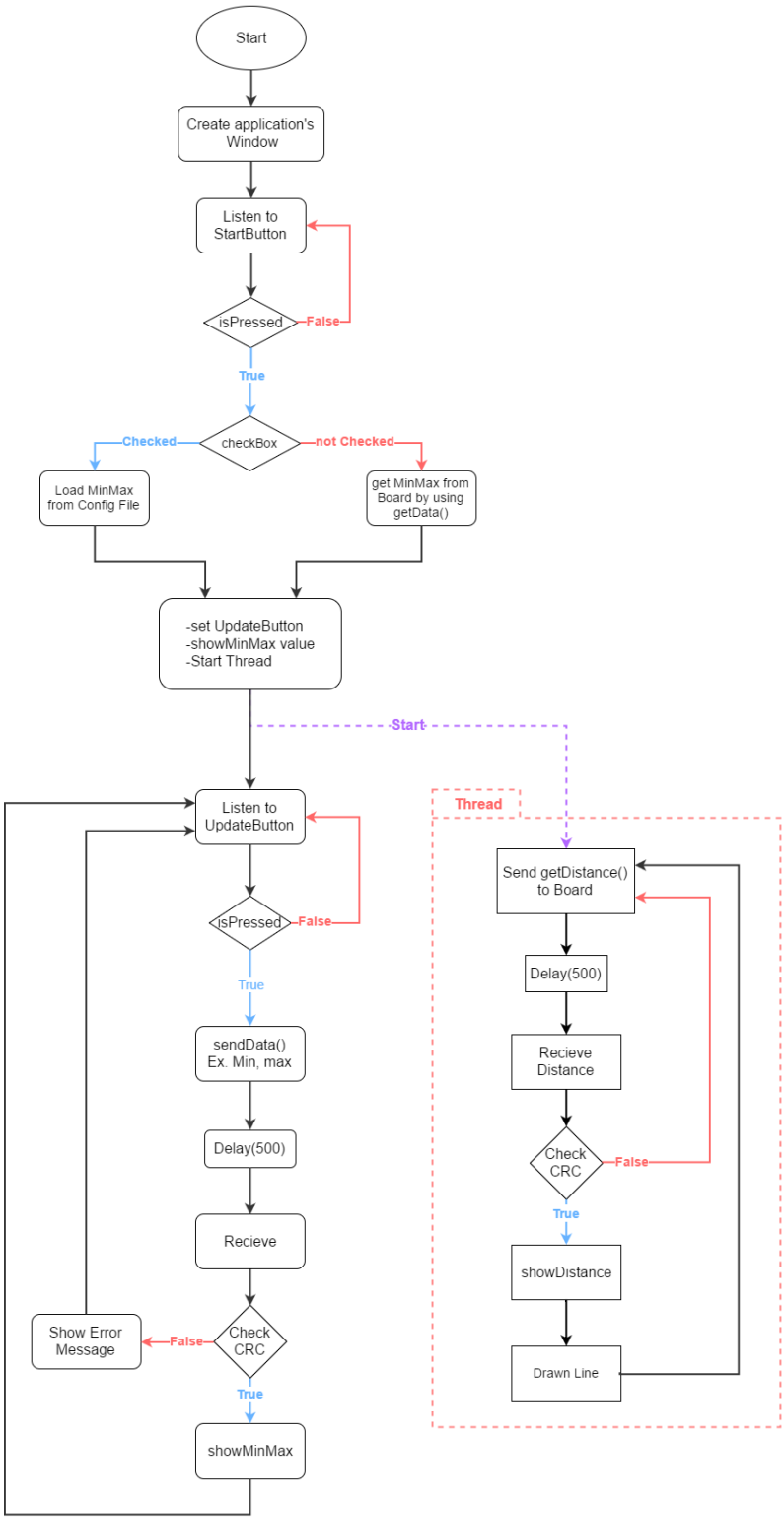
ภาพที่ 10: ภาพแสดงโครงสร้างการทำงานของโปรแกรมทางฝั่ง Arduino

จากภาพด้านบนโปรแกรมจะถูกแบ่งออกเป็นสองส่วน ได้แก่

ส่วนแรก ซึ่งทำหน้าที่ส่ง Request ไปหาเซ็นเซอร์ เพื่อขอรับค่า Distance ที่เซ็นเซอร์อ่านได้ ปละทำการบันทึกค่าระยะทางทั้งหมดนั้นลงในตัวแปล _distance และทำการเช็คเงื่อนไขว่าตรวจพบวัตถุหรือไม่เพื่อบันทึกลงในตัวแปล detected นั้นเอง

ส่วนที่สองทำหน้าที่รับคำสั่งจาก **Computer** และแปลความหมายเพื่อทำตามคำสั่งต่างๆ และเมื่อทำตามคำสั่งต่างๆเสร็จแล้วก็จะหน่วงเวลา **200 millisecond** จากนั้นก็จะวนกลับไปทำในส่วนแรกแบบนี้ไปเรื่อยๆ นั้นเอง

2. ฝั่ง Computer



ภาพที่ 11: ภาพแสดงโครงสร้างการทำงานของโปรแกรมทางฝั่ง Computer

จากภาพด้านบน เมื่อเริ่มทำงานโปรแกรมจะรอฟังคำสั่งเริ่มจากปุ่ม startButton เมื่อผู้ใช้กดปุ่ม ก็จะเช็คว่า checkbox ถูกเลือกหรือไม่ ถ้าถูกเลือกก็จะทำการอ่านค่า Min และ Max จากไฟล์ config.txt และส่งค่าไปยังบอร์ดเพื่อตั้งค่า Min และ Max แต่ถ้าไม่ได้เลือก ก็จะส่งคำขอไปยังบอร์ดเพื่อขอค่า Min และ Max มาแสดงผลนั้นเอง

เมื่อแสดงผลค่า Min และ Max เสร็จ โปรแกรมจะสร้าง Thread ขึ้นมา 1 Thread ซึ่งทำหน้าที่ส่งคำสั่ง getDistance() ไปยังบอร์ด เพื่ออัพเดทค่า Distance และวาดภาพบนหน้าจอทั้งหมด ทุกๆ 500 millisecond

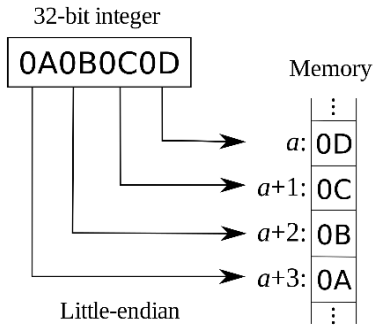
นอกจากสร้างThreadแล้ว ตัวโปรแกรมก็ยังต้องรอฟังคำสั่งจากปุ่ม “update” เพื่อทำการอัพเดทค่า Min และ Max ให้กับบอร์ดอีกด้วย

References

2 Byte Integer (Little-endian)

```
data = buffer[n] + (buffer[n+1] << 8); // [LO][HI]
```

เช่น (0x01, 0x05) = 1 + (5 * 256) = 1280



Bits

การระบุ Bit ในแต่ละ Byte ใช้การเรียงลำดับจากขวาไปซ้ายเป็นเลข 0 - 7

เช่น

Bit	7	6	5	4	3	2	1	0
Value	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**

CRC16

Effective data length < 16kB

```
unsigned char crc16_table[/*512*/] = {
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
1,192,128,65,0,193,129,64,0,193,129,64,1,192,128,65,
0,193,129,64,1,192,128,65,1,192,128,65,0,193,129,64,
0,192,193,1,195,3,2,194,198,6,7,199,5,197,196,4,
204,12,13,205,15,207,206,14,10,202,203,11,201,9,8,200,
216,24,25,217,27,219,218,26,30,222,223,31,221,29,28,220,
20,212,213,21,215,23,22,214,210,18,19,211,17,209,208,16,
240,48,49,241,51,243,242,50,54,246,247,55,245,53,52,244,
60,252,253,61,255,63,62,254,250,58,59,251,57,249,248,56,
40,232,233,41,235,43,42,234,238,46,47,239,45,237,236,44,
228,36,37,229,39,231,230,38,34,226,227,35,225,33,32,224,
160,96,97,161,99,163,162,98,102,166,167,103,165,101,100,164,
108,172,173,109,175,111,110,174,170,106,107,171,105,169,168,104,
120,184,185,121,187,123,122,186,190,126,127,191,125,189,188,124,
180,116,117,181,119,183,182,118,114,178,179,115,177,113,112,176,
80,144,145,81,147,83,82,146,150,86,87,151,85,149,148,84,
156,92,93,157,95,159,158,94,90,154,155,91,153,89,88,152,
136,72,73,137,75,139,138,74,78,142,143,79,141,77,76,140,
```

```
68,132,133,69,135,71,70,134,130,66,67,131,65,129,128,64
};

int crc16(unsigned char *data, int length)
{
        int index;
        int crc_Low = 255;
        int crc_High = 255;

        for (int i = 0; i<length; i++)
        {
                index = crc_High ^ (char)data[i];
                crc_High = crc_Low ^ crc16_table[index];
                crc_Low = (unsigned char)crc16_table[index+256];
        }

        return (crc_High << 8) + crc_Low;
}



byte _crc8(byte data[], int offset, int length)
{
    int crc = 0xFF;
    for (int i = 0; i < length; i++)
    {
        crc ^= data[i+offset];
        for (int b = 0; b < 8; b++)
        {
            if ((crc & 0x80) > 0) crc = (crc << 1) ^ 0x31;
            else crc = (crc << 1);
        }
    }
    return (byte)crc;
}
```

^ = XOR

การคำนวณ CRC16 จะคิดจากข้อมูลตัวแรกจนถึงข้อมูลก่อนหน้า CRC16 เช่น

ลำดับ	ชื่อ	ประเภทข้อมูล	ขนาด (byte)	รายละเอียด
0	Start		1	‘{‘
1	Command		1	
2 – 4	Data		3	
5	CRC16 (Hi)		1	
6	CRC16 (Lo)		1	
7	End		1	‘}’ }

** CRC16 ถูกคำนวณจากข้อมูลตำแหน่งที่ 0 – 4*

** CRC16 เรียง Hi ก่อน Lo (Big-endian)*

Revision History

Revision	Date	By	Change
0	2016-10-28	Supattra	Start
1	2016-11-08	Supattra	เพิ่ม valid และ detected