

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

【作業内容】

I2C インターフェースを付加した LCD をラズパイに接続して、文字列が表示できるようにプログラムを作成してください。

【作業項目】

1. I2C インターフェースの準備

i2cdetect -y 1 というコマンドを用いて、I2C デバイスの番号を調べる。結果、0x27 であることがわかった。

2. 下調べ

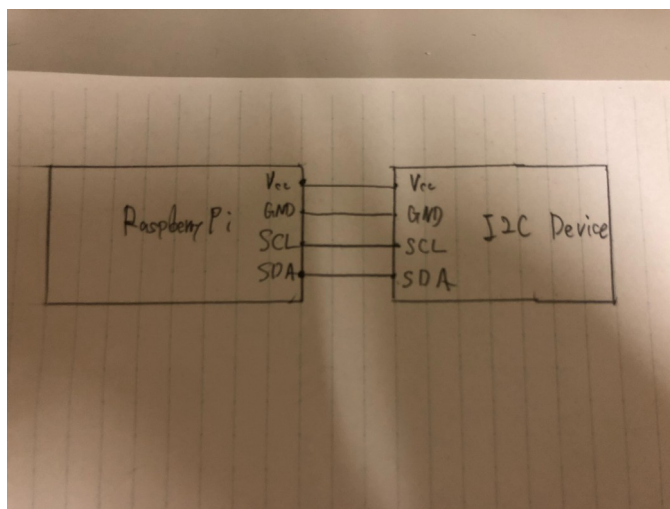
I2C は通信機器がマスターとスレーブになって通信をする。

SCL はクロック信号、SDL はデータ信号を通信しており、SCL が High のときの SDL の変化をみて、通信の開始と終了を調べている。

具体的には、SCL が High かつ SDL が立ち下がると通信開始。通信中に同じことが起こると通信終了となる。したがって、SDL でデータを通信するときは、SDL が Low のときのみ通信する。

3. 回路図

RaspberryPi の各ポートと I2C デバイスを接続する。回路図を以下に示す。



4. 任意の位置に文字を表示するのは時間の関係上できなかったため、先頭に A を表示するプログラムを示す。ただし、プログラム長が長いため、最後に乗せる。

作業時間】

- ・ 作業時間 : 200 分
- ・ 報告書作成時間 : 20 分

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

```
#include<wiringPi.h>
#include<wiringPiI2C.h>
#include<stdio.h>
#include<stdlib.h>

#define I2CReadMode 1
#define I2CWriteMode 0

/*
typedef struct {
    unsigned int bus4: 1;
    unsigned int bus5: 1;
    unsigned int bus6: 1;
    unsigned int bus7: 1;
} I2CDataBus;
*/

typedef struct {
    unsigned int register_selector: 1;
    unsigned int read_write: 1;
    unsigned int enable: 1;
    unsigned int back_light: 1;
    unsigned int bus4: 1;
    unsigned int bus5: 1;
    unsigned int bus6: 1;
    unsigned int bus7: 1;
} I2C8bitDataField;

typedef union {
    I2C8bitDataField field;
    u_int8_t raw_data;
} I2C8bitData;

I2C8bitData make_data_from_raw(u_int8_t data) {
    I2C8bitData ret = {.raw_data = data };
    return ret;
```

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

```
}
```

```
void show_i2c_data(I2C8bitData data) {
    printf("[DEBUG] I2C Data Details. %d\n", data.raw_data);
    if (data.field.register_selector == 1) {
        printf(" RS: Command Mode\n");
    } else {
        printf(" RS: Data Mode\n");
    }

    if (data.field.read_write == 1) {
        printf(" RW: Read Mode\n");
    } else {
        printf(" RW: Write Mode\n");
    }

    if (data.field.enable == 1) {
        printf(" Enable Bit: Enable\n");
    } else {
        printf(" Enable Bit: Disable\n");
    }

    if (data.field.back_light == 1) {
        printf(" Back Light: On\n");
    } else {
        printf(" Back Light: Off\n");
    }

    const u_int8_t transmission_data = (data.raw_data & 0xF0) >> 4;
    printf(" Transmission Data: %d, %d%d%d%d\n", transmission_data, (transmission_data & 0b1000)
    >> 3, (transmission_data & 0b0100) >> 2, (transmission_data & 0b0010) >> 1, transmission_data &
    0b0001);
}

u_int32_t raw_I2C_data_write(const u_int32_t fd, const I2C8bitData data) {
    const u_int32_t ret = wiringPiI2CWrite(fd, data.raw_data | 0b1000);
    if (ret != 0) {
```

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

```
printf("[ERROR] I2C write error. fd = %d, data = %d", fd, data);
return ret;
}
return 0;
}

void I2C_data_write(const u_int32_t fd, I2C8bitData data) {
    // show_i2c_data(data);

    data.field.enable = 0;
    raw_I2C_data_write(fd, data);
    delayMicroseconds(10);
    data.field.enable = 1;
    raw_I2C_data_write(fd, data);
    delayMicroseconds(10);
    data.field.enable = 0;
    raw_I2C_data_write(fd, data);
    delayMicroseconds(100);
}

void I2C_send_value(const u_int8_t fd, u_int8_t value, u_int8_t ctrl_option) {
    I2C8bitData data = {.raw_data = 0};

    // send high
    data.raw_data = (value & 0xf0) | ctrl_option;
    I2C_data_write(fd, data);

    // send low
    data.raw_data = ((value & 0x0f) << 4) | ctrl_option;
    I2C_data_write(fd, data);
}

void initialize_lcd(u_int32_t fd) {
    I2C8bitData data = {.raw_data = 0};

    I2C_data_write(fd, data);

    data.field.bus4 = 1;
```

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

```
data.field.bus5 = 1;
```

```
I2C_data_write(fd, data);
```

```
I2C_data_write(fd, data);
```

```
I2C_data_write(fd, data);
```

```
data.field.bus4 = 0;
```

```
I2C_data_write(fd, data);
```

```
// Transmission mode change
```

```
I2C_data_write(fd, data);
```

```
data.field.bus7 = 1;
```

```
data.field.bus5 = 0;
```

```
I2C_data_write(fd, data);
```

```
show_i2c_data(data);
```

```
/*
```

```
// Display off
```

```
data.raw_data = 0;
```

```
I2C_data_write(fd, data);
```

```
data.field.bus7 = 1;
```

```
I2C_data_write(fd, data);
```

```
// Display clear
```

```
data.raw_data = 0;
```

```
I2C_data_write(fd, data);
```

```
data.field.bus4 = 1;
```

```
I2C_data_write(fd, data);
```

```
*/
```

```
// Entry mode
```

```
data.raw_data = 0;
```

```
I2C_data_write(fd, data);
```

```
data.field.bus6 = 1;
```

```
data.field.bus5 = 1;
```

```
I2C_data_write(fd, data);
```

作業報告書（2020 年 10 月 5 日）

j18404 岩橋涼介

```
show_i2c_data(data);

}

int main() {
    u_int32_t fd = wiringPiI2CSetup(0x27);
    printf("[INFO] Start I2C Connection. fd = %d\n", fd);

    initialize_lcd(fd);

    I2C_send_value(fd, 'A', 1);
}
```