

作業報告書 (2022 年 1 月 30 日)

J20413 北野正樹

【作業内容】

プログラムの理解と作成

【作業項目】

① wiringPi のライブラリを使ったプログラミング

wiringPi の SPI 用関数は、初期化関数として wiringPiSPISetup(ch, speed)、データの送受信関数として wiringPiSPIDataRW(ch, data, byte)を用いる。

・ int wiringPiSPISetup(int ch, int speed)

Ch: 0 | 1 チャンネルを与える。

0 は CE0 (BCM_GPIO8)、1 は CE1 (BCM_GPIO7) と連動するので注意。

指定したチャンネルの CE0 または CE 1 がアクセス時に 0 となる。

CE0 を CS/SHDN に接続したなら、0 を指定することになる。

Speed: SPI 通信のクロック周波数は Hz で指定する。32Mhz まで対応可能。

MCP3008 では、10k Hz が 10 ビット A/D 変換時の最大周波数とされている。よって 10000 を与えることとなる。

戻り値: 成功時はファイルディスクリプタ、失敗時は-1

・ int wiringPiSPIDataRW(int ch, unsigned char *data, int len)

Ch: 0 | 1 スレーブのチャンネルを与える。上と同じ。

Data: 送受信するデータのバッファとなる配列名を指定する。

なお、送受信終了時に、受信データによって送信データは上書きされる。

実行前: data[0]="0000 0001", data[1]="SCCC XXXX", data[2]="XXXX XXXX"

➔ 例) S: 1 (シングルエンド指定)、CCC:"000" (ch0 を指定)

※X は 0 でも 1 でもよいことを示す。

実行後: data[0]="???? ????", data[1]="???? ?0BB", data[2]="BBBB BBBB"

➔ Data[1]の回 2 ビットを data[2]の 8 ビット上位ビットとして連結

※?はそのビットの値が不定であることを示す。

Len: 3 送受信するデータの長さ (バイトで指定) を指定する。

戻り値: 不明 (記述がない)

なお、wiringPiSPIDataRW関数を使わなくても、システムコール read(), write()でもアクセス可能である。データ読み込みの必要がない、カラーLED やシフトレジスタなどのアクセスには write()を使うのが良い方法である。ただし、今回は A/D 変換器に書き込みながら読み出しが必要なので、この関数を使用する。

【作業時間】

・ 作業時間: 60 分

・ 報告書作成時間: 60 分