

Курсовая работа

Задача.

В качестве проверки знаний, усвоенных в процессе освоения курса, предлагается организовать межпроцессорную передачу данных по интерфейсу SPI между платой Arduino и отладочной платой Nucleo.

1. Необходимо определить нуждается ли соединение двух отладочных плат в преобразовании уровней логических сигналов. При необходимости, предложить и реализовать схему такого подключения.
2. Передать данные из платы Arduino (информацию о напряжении или освещенности) в отладочную плату Nucleo.

Аппаратная часть.

Анализ документации.

Известно, что рабочим напряжением отладочной платы Arduino Uno является 5 В.

Power	Circuit operating voltage	5V
	Input voltage (recommended)	7-12V
	Input voltage (limit)	6-20V
	Supported battery	9V battery
	Battery connector	Power jack
	DC Current per I/O Pin	20 mA
	DC Current for 3.3V Pin	50 mA

Ист. Документация на Arduino uno - <https://docs.arduino.cc/hardware/uno-rev3>

Рабочее напряжение микроконтроллера STM32F103RB, установленного на отладочную плату STM32 Nucleo-64 board, является 3.3 В.

Table 6. Voltage characteristics				
Symbol	Ratings	Min	Max	Unit
$V_{DD} - V_{SS}$	External main supply voltage (including V_{DDA} and V_{DD}) ⁽¹⁾	-0.3	4.0	V
V_{IN} ⁽²⁾	Input voltage on five volt tolerant pin	$V_{SS} - 0.3$	$V_{DD} + 4.0$	
	Input voltage on any other pin	$V_{SS} - 0.3$	4.0	
$ \Delta V_{DDx} $	Variations between different V_{DD} power pins	-	50	mV
$ V_{SSx} - V_{SS} $	Variations between all the different ground pins	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see Section 5.3.11: Absolute maximum ratings (electrical sensitivity)		

Ист. Документация к микроконтроллеру STM32F103xB.

Таким образом, для совместной работы и организации межпроцессорной передачи данных по интерфейсу SPI между платами Arduino Uno и Nucleo необходимо согласование логических уровней 5В и 3.3В

Такое согласование может быть выполнено различными способами, например, простейшим делителем напряжения, микросхемой с соответствующим функционалом и т.д.

Вместе с тем, согласно документации, микроконтроллер STM32F103xB поддерживает (толерантен) подключение некоторых выводов по линии питания 5 В (кол. FT).

J8	K12	26	G8	34	52	-	PB13	I/O	FT	PB13	SPI2_SCK/ USART3_CTS ⁽⁹⁾ / TIM1_CH1N ⁽⁹⁾	-
H8	K11	27	F8	35	53	-	PB14	I/O	FT	PB14	SPI2_MISO/ USART3_RTS ⁽⁹⁾ / TIM1_CH2N ⁽⁹⁾	-
G8	K10	28	F7	36	54	-	PB15	I/O	FT	PB15	SPI2_MOSI/ TIM1_CH3N ⁽⁹⁾	-

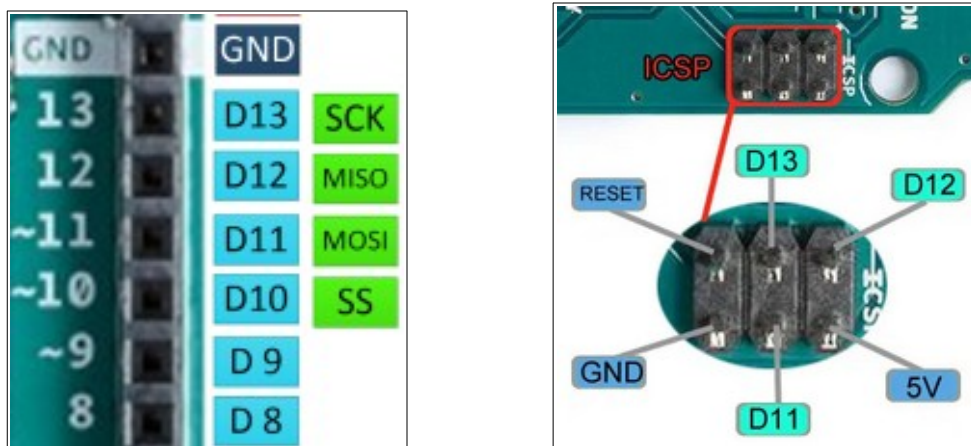
Ист. Документация к микроконтроллеру STM32F103xB.

Как видно из таблицы, выводы PB13, PB14 и PB15 поддерживают возможность их подключения к 5 В, при этом они являются выводами, отвечающими за интерфейс SPI.

Таким образом, имеется возможность соединения двух плат без использования дополнительных компонентов (также вероятно безопасней).

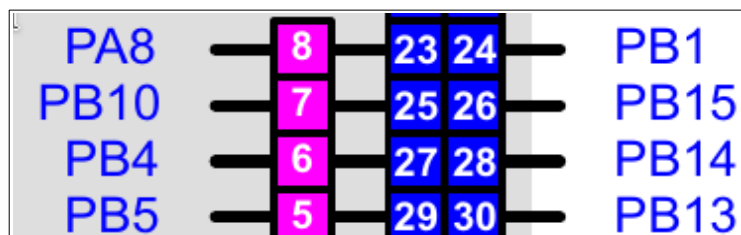
Подключение отладочных плат.

На плате Arduino Uno за интерфейс SPI отвечают выводы D10, D11, D12 и D13, назначенные на SS, MOSI, MISO, и SCK соответственно. А также блок контактов ICSP.



Выводы SPI на плате Arduino Uno.

На плате Nucleo за интерфейс SPI отвечают выводы PB13, PB14 и PB15, назначенные на SCK, MISO и MOSI соответственно.



Выводы SPI на плате Nucleo.

Выводы одной платы подключаются к соответствующим одноименным на другой плате.

Получение данных об освещении.

Для получения данных об освещенности используется делитель напряжения, состоящий из фоторезистора, который подключен как одно из плечей. В качестве второго плеча был выбран резистор номиналом 1 кОм.

Подключение фоторезистора произведено к аналоговому выводу A0 платы Arduino Uno.

Программная часть.

Arduino.

Далее приведен полный исходный код прошивки для arduino.

```
#include <Arduino.h>
#include <SPI.h>
#include <inttypes.h>

uint8_t data = 0;

void setup()
{
    SPI.begin();
    SPI.setClockDivider(SPI_CLOCK_DIV2);
    digitalWrite(SS, LOW);
}

void loop()
{
    /* для передачи байта конвертируем полученные данные в 8-ми битный размер. */
    data = map(analogRead(A0), 0, 1023, 0, 255);
    SPI.transfer(data);
    delay(1000);
}
```

Nucleo.

Далее приведена часть исходного кода прошивки для STM32, отвечающая за получение данных.

```
while (1) {
    HAL_SPI_Receive(&hspi2, &in_buffer, sizeof(in_buffer), 1000);
    printf("data: %d\n", in_buffer);
}
```

Результаты.

Ниже приведены результаты межпроцессорной передачи данных об освещении из платы Arduino в отладочную плату Nucleo. Данные получены с использованием serial port (Arduino) и отладочного режима среды разработки CubeIDE (STM32).

```
20
19
20
172
172
169
169
165
20
19
19
19
19
19
19
U:~%- /dev/ttyACM0 75% (132,0) (Term: char 9600 8N1 open yas MMap)
```

Внутренние данные платы Arduino, выведенные в терминал (среда Emacs).

```
Problems Console Executables Debugger Console Memory SWV ITM Data Console
Port 0
data: 20
data: 20
data: 19
data: 19
data: 20
data: 172
data: 172
data: 169
data: 169
data: 165
data: 20
data: 19
```

Данные полученные платой Nucleo и выведенные ею в терминал (CubeIDE).

Как видно из приведенных изображений, данные переданы и получены корректно, за исключением одного значения (задвоилось значение 19). Предполагается, что низкий уровень значений показывает низкую освещенность, высокий соответственно высокую освещенность. Типичное значение освещенности в помещении в ходе работы устройств 20 (комнатный свет), при подсветке фонарем показатель освещенности поднимался до значения 165 — 172 с учетом того, что диапазон освещенности от 0 до 255 (8 бит).