**Lab 4: Interfacing Sensors via SPI**

**Instructor: Prof. Yifeng Zhu**

**Fall 2019**

## Goals:

1. Understanding the SPI configuration used by the STEVAL-FCU001V1
2. Initializing an SPI interface to communicate with the LSM6DSL, LPS22HD, and LIS2MDL sensors present on the STEVAL-FCU001V1

## Summary of Tasks:

1. Read Chapter 22.3 Serial Peripheral Interface Bus (SPI)
2. Programming the SPI peripheral
   1. Initialize the SPI2 interface and its associated GPIO pins by implementing DrvStatusTypeDef(), Sensor\_IO\_SPI\_CS\_Init\_All(), and Sensor\_IO\_SPI\_CS\_Init().
   2. Verify the initialization using the smartphone application to request and collect data

**Background:**

This lab focuses on the SPI interface used by the STEVAL to interface with the various sensors present on the device. Specifically, the STEVAL uses the SPI2 interface of the STM32F4 to interface with three sensors: LSM6DSL (accelerometer and gyroscope), LPS22HD (pressure sensor), and LIS2MDL (magnetometer). In addition, it uses SPI1 to communicate with the Bluetooth module (SPBTLE-RF).

Figure 1. The microcontroller uses SPI2 to interface three sensors

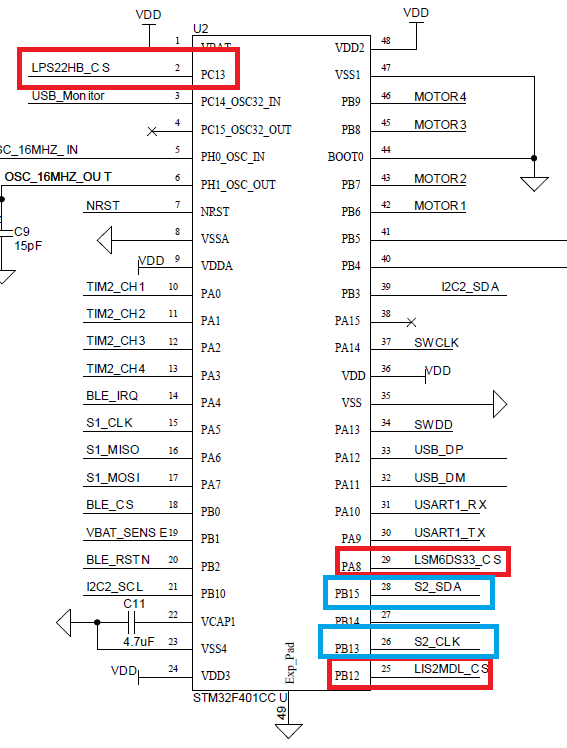
When interfacing with these sensors, the STEVAL broadcasts and receives data over a single pin, connected to all three of the sensors. To allow for proper communication using this single pin, each sensor is controlled via a separate CS pin. Each sensor only listens to incoming commands when its CS pin is held low. By changing the values of the CS pins, the STEVAL is able to select which sensor receives each command it sends. A diagram on the following page depicts a schematic of the STEVAL, with pins relevant to this lab highlighted.

The three red highlighted pins, PA8, PB12, and PC13, are the CS pins for the LSM6, the LIS2, and the LPS22, respectively, while the two blue highlighted pins, PB13 and PB15, are the clock and data lines used by the SPI2 interface. The configuration of these pins, as well as the SPI2 interface itself, will be the focus of this lab. In this lab, the three red pins will simply be configured as outputs, while the two blue pins will be configured to perform an Alternate Function (AF), namely, to act as the clock and data lines of the SPI interface, rather than as GPIOs.

|  |  |  |
| --- | --- | --- |
| Pin | Function | Comment |
| PB 13 | SPI2\_CLK |  |
| PB 15 | SPI2\_SDA |  |
| PC 13 | LPS22HB\_CS | Chip select |
| PA 8 | LSM6DS33\_CS | Chip select |
| PB 12 | LIS2MDL\_CS | Chip select |

The SPI2 interface will be configured as a master, to control the attached sensors. As only a single line is used to transmit data, the 1-line bidirectional data mode will be used. This mode has the SPI interface switch between transmitting and receiving data on the same line. The data frame format, clock polarity, data capture edge, and bit transmission order settings will all be set according to the requirements of the sensors. An acceptable baud rate will also be chosen according to the sensor’s requirements and capabilities. TI mode will be disabled to allow for many of these settings to be modified.

First, complete the pre-lab to determine the bit masks required to set these parameters, among others. After completing the pre-lab, follow the lab procedure to configure the SPI driver and GPIOs using the provided code. After configuring the device, test your code by checking the data output of the drone app. The operation of the drone app is illustrated in Appendix A.



**Pre-Lab Assignment**

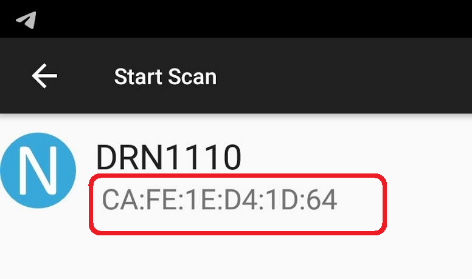
**Lab 4: Interfacing Sensors via SPI**

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Time & Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Before you come to lab find a space where you would not have a lot of other people not using their drone’s. Program the board with the Fly code. Next, open the ST Drone app and connect to one of the drone’s listed, then look at the gyro and accelerometer data and move your board around and see if that correlates. If it does, write down the BLE address listed below the Name. This is your drone’s unique address.



1. **Configure SPI Peripheral Registers**

Configure the SPI with the following parameters:

Master mode enabled, slave select pin set high, 1-line bidirectional data mode, 8-bit data frame format, clock polarity high, first data capture edge on 2nd clock transition, software slave management enabled, SS output enabled, baud rate set at 1/16 f\_PCLK, MSB transmitted first, TI mode disabled, CRC calculations disabled.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Offset** | **Register** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0x00 | **SPIx\_CR1** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | BIDIMODE | BIDIOE | CRCEN | CRCNEXT | DFF | RXONLY | SSM | SSI | LSBFIRST | SPE | BR[2:0] | | | MSTR | CPOL | CPHA |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x04 | **SPIx\_CR2** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | LDMA\_TX | LDMA\_RX | FRXTH | DS[3:0] | | | | TXEIE | RXNEIE | ERRIE | FRF | NSSP | *SSOE* | TXDMAEN | RXDMAEN |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x08 | **SPIx\_SR** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | FTLVL[1:0] | | FRLVL[1:0] | | FRE | BSY | OVR | MODF | CRCERR | Res. | Res. | TXE | RXNE |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x0C | **SPIx\_DR** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | DR[15:0] | | | | | | | | | | | | | | | |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x10 | **SPIx\_CRCPR** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | CRCPOLY[15:0] | | | | | | | | | | | | | | | |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x14 | **SPIx\_RXCRCR** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | RxCRC[15:0] | | | | | | | | | | | | | | | |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x18 | **SPIx\_TXCRCR** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | TxCRC[15:0] | | | | | | | | | | | | | | | |
| Value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Lab Procedure**

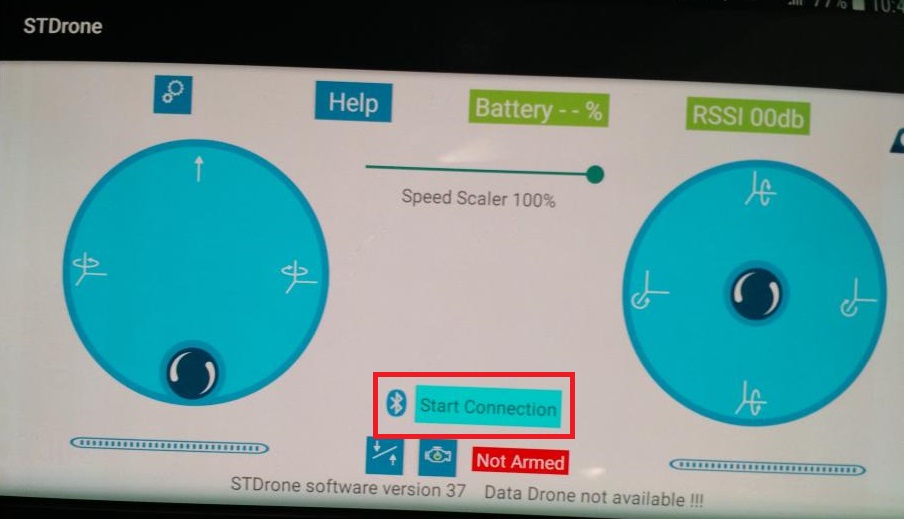
Perform the following:

1. In the **Sensor\_IO\_SPI\_Init()** function of the steval\_fcu001\_v1.c file within the drone project folder, add the following code snippets:
   1. Code to enable the clock to SPI2 and GPIOB
   2. Code to configure PB13 and PB15 to be:
      1. AF mode
      2. Push-pull
      3. No pull-up, no pull-down (NO PUPD)
      4. High speed
      5. Using Alternation Function 5 (AF5)
   3. Code to configure SPI2 as done in the pre-lab
   4. Code to prepare SPI2 for transmission by:
      1. Setting the bidirectional transfer pin to output mode
      2. Enabling SPI2
2. In the Sensor\_IO\_SPI\_CS\_Init\_All() function of the same file within the drone project folder, add the following code snippets:
   1. Code to enable the clock to GPIOA, GPIOB, and GPIOC
   2. Code to set the output of PA8, PB12, and PC13 high
   3. Code to configure PA8, PB12, and PC13 to be:
      1. Configured as an output
      2. High speed
      3. NO PUPD
3. Some calls to the SPI driver require only a single device’s pin to be initialized. Modify the Sensor\_IO\_SPI\_CS\_Init() function similarly to the function in the previous step:
   1. Code to enable GPIOA, set PA8, and configure PA8 as in Step 2c
   2. Code to enable GPIOB, set PB12, and configure PB12 as in Step 2c
   3. Code to enable GPIOC, set PC13, and configure PC13 as in Step 2c
4. Observe sensor data in smart phone Apps
   1. Once these steps are complete, compile the code and connect to the board using the smartphone application. In the application’s settings menu, enable the setting that allows for the sensor data to be seen. If the first two steps were completed correctly, correct data should be output there. A pressure reading around 1000mBar (approximate atmospheric pressure) should be shown, and outputs from the gyroscope and accelerometer should change as the board is moved and tilted. Verify correct data output with your TA, who will then check your code and mark you off for this lab.

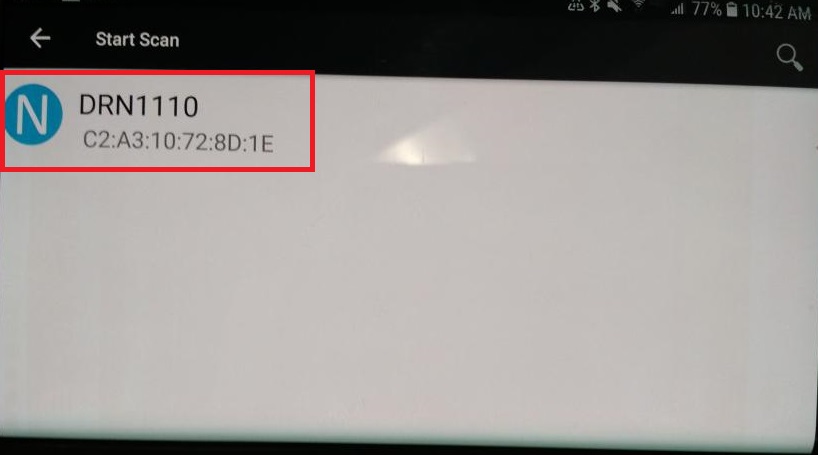
**Appendix A: Drone App Usage**

The Android app can be downloaded from Google play: <https://www.st.com/en/embedded-software/appdrone.html>

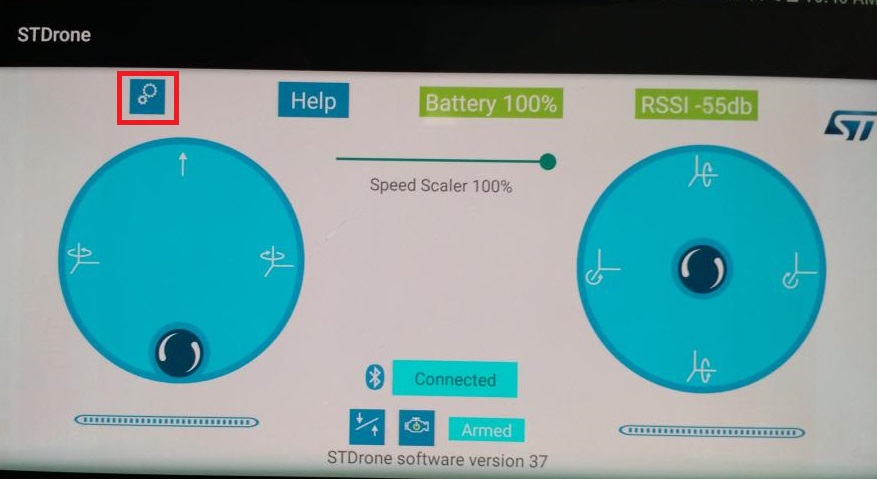
This appendix will explain how data can be collected from the STEVAL by using the STDrone smartphone application. First, open the application, then press the “Start Connection” button, highlighted below, to connect to your STEVAL.



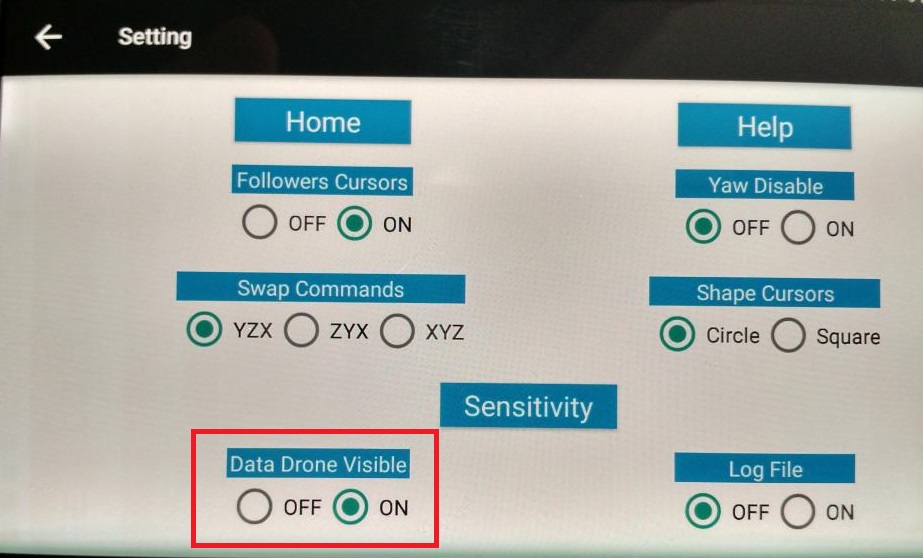
This button will move you to a different screen. Select your device from among those listed. If you are not sure which device is yours, try having other lab members temporarily power off their STEVALs, so that only your board appears onscreen.



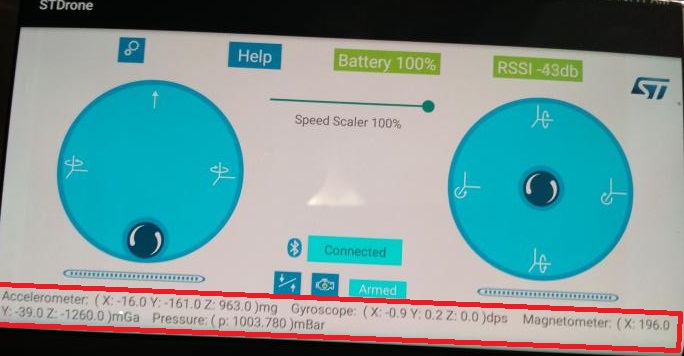
Now, return to the main screen, and press the settings button, as indicated in the below image.



On the settings screen, turn the “Data Drone Visible” setting ON, as indicated below.



Now, after returning again to the main screen, you should see a collection of data along the bottom of the screen.



The values shown for the accelerometer and gyroscope should change as you move and tilt your device. An easy way to determine whether the correct data is being output by the device is to examine the pressure reading. Typically, the atmospheric pressure at sea-level is approximately 1013 mBar, so a value ~1000mBar, such as the 1003mBar in the above image, indicates a correct reading. Because the code written in this lab is used by all of the sensors, and the SPI interface for each sensor is initialized in the same way, a correct reading from the pressure sensor is a strong indicator for correct readings from the other sensors connected through SPI2.