Lorhea Grace M. Villena
BET-COET-3A

MODULES THAT USES I2C

OLED Displays (128x64 or 128x32)

Organic light-emitting diode Displays use the I2C protocol to send data from a microcontroller for rendering text or graphics. The OLED connects via two data lines, SDA and SCL, with the microcontroller acting as the I2C master and the display as the slave. Each I2C device has a unique address, enabling communication over just two wires, making it ideal for projects with limited pin availability.

Real-Time Clocks (RTC)

The **DS1307** and **DS3231** RTC modules use I2C to communicate with a microcontroller by connecting through SDA and SCL lines. The microcontroller sends or receives time and date data via these lines. While the DS1307 is less accurate, the DS3231 has a built-in temperature-compensated oscillator for higher precision. Both modules maintain time even during power loss with a backup battery.

3. Temperature & Humidity Sensors

- The **HTU21D** and **SHT31** sensors use the I2C protocol to communicate with a microcontroller, sending temperature and humidity data. They connect via two lines, SDA and SCL, with the microcontroller acting as the master and the sensors as slaves. The sensors have unique I2C addresses, and the microcontroller requests or receives data from them over these two wires, enabling efficient and simple communication.
- The **BMP280** and **BME280** sensors use the I2C protocol to communicate with a microcontroller, providing temperature, pressure, and in the case of the BME280, humidity data. They connect through SDA and SCL lines, with the microcontroller as the master and the sensors as slaves. The microcontroller uses the sensors' I2C addresses to send commands and retrieve data over the two-wire connection for efficient communication.

4. EEPROM (Electrically Erasable Programmable Read-Only Memory)

The **AT24C32** EEPROM uses the I2C protocol to store and retrieve data via two lines, SDA and SCL. The microcontroller acts as the I2C master, communicating with the AT24C32, which has a unique I2C address. Data is written to or read from the EEPROM's memory cells over these two wires, allowing for non-volatile data storage in embedded systems.

5. Light Sensors

The **BH1750** light sensor uses the I2C protocol to communicate with a microcontroller, enabling it to measure ambient light levels. It connects via SDA and SCL lines, with the microcontroller as the master and the BH1750 as the slave. The microcontroller sends commands to the sensor using its unique I2C address, and in response, it receives light intensity data, allowing for straightforward integration into various projects.
The **TSL2561** light sensor uses the I2C protocol to communicate with a microcontroller by connecting through the SDA and SCL lines. The microcontroller acts as the I2C

master, sending commands to the sensor, which serves as the slave with a unique address. This setup allows the microcontroller to request ambient light data efficiently over just two wires, making it suitable for various applications.

6. Digital Accelerometers

The **ADXL345** and **MPU6050** sensors utilize the I2C protocol to communicate with a microcontroller, providing acceleration and motion data. They connect through the SDA and SCL lines, with the microcontroller serving as the I2C master and the sensors as slaves. Each sensor has a unique I2C address, allowing the microcontroller to request or read data efficiently over just two wires.

7. Proximity and Gesture Sensors

The **APDS-9960** sensor uses the I2C protocol to communicate with a microcontroller for gesture recognition, proximity detection, and ambient light sensing. It connects via SDA and SCL lines, with the microcontroller as the master and the APDS-9960 as the slave. The sensor has a unique I2C address, allowing the microcontroller to request data or configure settings over the two-wire interface, facilitating efficient communication in compact designs.

8. PWM Drivers

The **PCA9685** uses the I2C protocol to control multiple PWM outputs for applications like servo and LED control. It connects to a microcontroller via two lines, SDA and SCL, where the microcontroller serves as the I2C master and the PCA9685 as the slave. By sending commands to the PCA9685's unique I2C address, the microcontroller can configure PWM settings and manage up to 16 channels efficiently using just two wires.

9. Current and Voltage Sensors

The **INA219** and **INA260** sensors use the I2C protocol to measure and communicate voltage, current, and power data to a microcontroller. They connect via the SDA and SCL lines, with the microcontroller serving as the master and the sensors as slaves. Each sensor has a unique I2C address, allowing the microcontroller to request measurements and receive data efficiently over the two-wire interface.

10. Magnetometer and Compass Modules

The **HMC5883L** is a digital compass that uses the I2C protocol to communicate with a microcontroller, allowing it to measure magnetic fields in three axes. It connects via SDA and SCL lines, with the microcontroller as the master and the HMC5883L as the slave. The sensor has a unique I2C address, enabling the microcontroller to request magnetic data efficiently over just two wires, making it suitable for navigation and orientation applications.

MODULES THAT USES SPI

They connect via four main pins: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), and CS (Chip Select).

1. SD Card Modules

SD Card modules use the SPI protocol to enable communication between a microcontroller and the SD card for reading and writing data. The microcontroller acts as the SPI master, sending commands to the SD card, which responds with data. This efficient communication method allows for fast data transfer rates, making it ideal for applications that require storage and retrieval of large amounts of data.

2. OLED and TFT Displays

The **ST7735** and **ILI9341** are TFT display controllers that use the SPI protocol to communicate with microcontrollers for rendering graphics and text. The microcontroller acts as the master, sending commands and pixel data to the displays, which act as slaves. This high-speed communication allows for efficient updates and rendering of images, making these displays suitable for a variety of embedded applications.

3. Flash Memory Chips

The **W25Q32** and other serial NOR flash memory chips use the SPI protocol to communicate with a microcontroller, enabling fast data storage and retrieval. The microcontroller acts as the master, sending commands and data to the flash memory, which responds with the requested data. This high-speed interface allows for efficient read, write, and erase operations, making it ideal for applications requiring non-volatile storage.

4. Accelerometers and Gyroscopes

The **MPU6050** and **L3GD20** are motion sensors that use the SPI protocol to communicate with a microcontroller, providing acceleration and gyroscope data. The microcontroller acts as the master and the sensors as slaves. The sensors respond to specific commands sent by the microcontroller, enabling efficient data exchange and real-time motion tracking for various applications.

5. Digital Potentiometers

The **MCP4131** is a digital potentiometer that uses the SPI protocol to communicate with a microcontroller for adjusting resistance values. The microcontroller acts as the SPI master, sending commands and data to the MCP4131 to set the desired resistance, allowing for precise control in various applications like signal processing and sensor calibration.

6. RF Modules

The **NRF24L01** and **ESP8266** modules use the SPI protocol for high-speed wireless communication with microcontrollers. The NRF24L01 communicates wirelessly for data transmission, while the ESP8266, primarily a Wi-Fi module, can connect to the internet and transfer data using SPI. This protocol enables efficient and fast data exchange between the microcontroller and the modules, suitable for various wireless applications.

7. ADC (Analog-to-Digital Converters)

The **MCP3008** and **ADS1115** are analog-to-digital converters (ADCs) that use the SPI protocol to communicate with a microcontroller. The microcontroller acts as the master, sending commands to the ADC to initiate conversions and receive digital values for the corresponding analog inputs. This high-speed communication allows for efficient sampling of multiple analog signals, making these ADCs suitable for various sensor applications.

8. Real-Time Clock (RTC) Modules

The **DS3234** is a real-time clock (RTC) module that uses the SPI protocol to communicate with a microcontroller. The microcontroller acts as the SPI master, sending commands and receiving time data from the DS3234, which acts as the slave. This high-speed communication allows for precise timekeeping and configuration of the RTC while minimizing pin usage.

9. Temperature Sensors

The **MAX6675** and **MCP9808** temperature sensors use the SPI protocol to communicate with a microcontroller. The microcontroller acts as the master, sending a signal to select the sensor and requesting temperature data, which is transmitted back through the MISO line. This allows for fast and efficient data transfer, making them suitable for accurate temperature measurement in various applications.

REFERENCES:

- Santos, S., & Santos, S. (2023, April 23). *Guide for I2C OLED Display with Arduino | Random Nerd Tutorials*. Random Nerd Tutorials. https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/
- LME Editorial Staff. (2022, October 10). *Interface DS1307 RTC Module with Arduino*. Last Minute Engineers. https://lastminuteengineers.com/ds1307-rtc-arduino-tutorial/
- Industries, A. (n.d.). *AdaFruit HTU21D-F Temperature & Humidity Sensor Breakout Board*. https://www.adafruit.com/product/1899
- *High-precision RTC module DS3231 AT24C32 I2C Module (without battery) | Shop at Bbiri Centre*. (n.d.). Shop at Bbiri Centre. https://www.bbiri-centre.com/product-page/real-time-clock-rtc
- Santos, S., & Santos, S. (2022, March 11). *Arduino with BH1750 Ambient Light Sensor | Random Nerd Tutorials*. Random Nerd Tutorials. https://randomnerdtutorials.com/arduino-bh1750-ambient-light-sensor/
- SPI with DMA and TFT Display ST7735. (2024, May 21). https://community.st.com/t5/stm32-mcus-embedded-software/spi-with-dma-and-tft-display-st7735/td-p/674378
- ISP programming W25Q32 SPI flash memory chip. (n.d.). Electrical Engineering Stack Exchange. https://electronics.stackexchange.com/questions/484889/isp-programming-w25q32-spi-flash-memory-chip
- MarkSherstan. (n.d.). GitHub - MarkSherstan/STM32-MPU6050-MPU9250-I2C-SPI: STM32 MPU6050/9250 I2C and SPI interface. Sensor fusion using a complementary filter yields sensor Euler angles and is implemented in both C and CPP. GitHub. https://github.com/MarkSherstan/STM32-MPU6050-MPU9250-I2C-SPI
- Instructables. (2023, February 14). Arduino with MCP4131 Digitally Controlled Potentiometer (DCP). Instructables. https://www.instructables.com/Arduino-With-MCP4131-Digitally-Controlled-Potentio/
- MAX6675 with hardware SPI - Arduino Reference. (n.d.). https://reference.arduino.cc/reference/en/libraries/max6675-with-hardware-spi/