Lesson 5
Programming Arduino Examples 9.5 and 9.6

Example 9.5

- Serial UART data read of a RFID Integrated circuit (IC) at an Arduino board using UART protocol using Tx and Rx pins

Example 9.6

- Use of Serial Soft library-functions at IDE for reading I2C bus data
- Serial I2C bus data and calibration coefficients read from IC based sensor

Example 9.5 RFID tag serial output reading using UART serial input

- A byte represents a character in a string, a sensed data, and command for receiving device or address of destined or receiving device
- UART protocol based communication two signals
- Tx or TxD (for serial reception) and Rx or RxD (for serial reception)

Example 9.5 RFID tag serial output reading using UART serial input

- When set baud rate 2400
- Then, 240 times a new set of 8-bits data transmits in one second.
- 10 bits transmit for each set for data, character, command or address

Example 9.5 RFID tag serial output reading using UART serial input

- Arduino board has pins 0 (Rx) and 1 (Tx) for UART serial communication. Since a computer connects to the board using these pins.
- Software Serial library has functions, which read and write serial data using Arduino board
- Thus, let pins 3 and pin 4 as the Rx and Tx, respectively.
- Declare int SerialRxPin 3 int SerialTxPin 4

Example 9.5 RFID Tag UART Output pin Reading Using Serial Input

- Let the RFID IC send 12 bytes, one for the header (address), ten bytes for the sensed data or RFID tag or ID and one byte for the end character. (Data sheet of the device specifies the actual address and format of header bits and tag or ID bits.)*/

Example 9.5 RFID Tag UART Output pin Reading Using Serial Input

- #include <UARTSerial.h> /*Serial IO functions for Arduino UART Serial Interface*/
- Assume that RFID IC uses address 0x08 for device header which includes address.
- #define UARTDEVICE_HEADER 0x08
- int nID 10 /* number of characters in the string for the ID or number of bytes of sensed data from the device*/

Example 9.5

- Declaring method for requesting Rx data
- SerialRxRequest : : SerialRxRequest (bool serialRx_request)

Setup ( )

- Setup pin modes
- pinmode (SerialRxPin, INPUT);
- pinmode (SerialTxPin, OUTPUT);
- pinMode (internalLED, OUTPUT);
- digitalWrite (internalLED, HIGH);
- digitalWrite (SerialTxPin, LOW);//enable Serial reception

loop ( )

- If (SerialTxPin = = LOW || SerialRx_request = = true) {
- SerialRx.begin (4800); /*Assume it communicates at baud rate = 4800. */
- ID_characters [0] = SerialRx.read (); /* Reader Header character
- /*Check header character = 0x08, if match then read sensed data ten bytes and end character*/

Example 9.6

- I2C protocol a serial-bus protocol that communicates in synchronous mode.
- Device transfers data as master or slave.
- Master means the device can address and communicate with a number of slaves. Master sends clock pulses to the slave devices for synchronisation. Maximum connected devices can be up to 127.

Example 9.6

- At one instance, one device functions as master and another as slave, and functioning depends on the internal circuit for I2C functions at the devic.
- Assume that a sensor has built-in ADC as well as I2C interface, and communicates the bytes over I2C SDA and SCL pins.

Example 9.6

- Read the data and coefficients from IC based sensor at Arduino serial-port
- Programming of Arduino for usages of synchronous serial-data read using I2C bus
- The serial pin A4 is for SDA bits and A5 for SCL bits.
- Assume Arduino functions as master device and an IC based sensor as a slave device

Example 9.6

- Assume that the sensor IC saves addresses for control and data registers. The registers save at EEPROM memory in the IC.
- A measured 16-bit value saves at two memory addresses, which save the data registers
- A programming of control register enables the working of the IC in several modes. The IC uses the data registers and communicates two bytes.

Example 9.6

- Two bytes: One is for higher and other for lower bytes.
- Memory also saves and communicates the calibration coefficients for communication as I2C serial data from a set of addresses
- Assume device calibration data has 22 addresses in memory for eleven values of 16-bit calibration coefficients.

Example 9.6

- Declaring the data types, constants, variables and functions used.
- #include <Serial.h> /* Use Serial Monitor*/
- #include <Wire.h> /* include Wiring functions*/
- #include <util.h> /*IO utility*/

Define Register and Memory Addresses at Sensor IC

- #define I2CDEVICE_ADDRESS 0x08
- /* control command address 0x02*/
- #define char I2DEVICE_CONTROLDATA 0x02
- /*Assume Data 1 address is at 0x06 and 2 at 0x0A. */
- #define I2DEVICE_DATA1_ADDRESS 0x06
- #define I2DEVICE_DATA2_ADDRESS 0x0A
- #define CALIBCOEFF_ADDRESS [ ] 0x18, 0x1A, 0x1C, 0x1E, 0x20, 0x22, 0x24, 0x26, 0x28, 0x2A, 0x2C

Setup ( )

- void **setup ( ) {**
- int observedValue0, observedCValue0,
- Serial.begin (9600); //Let UART mode baud rate = 9600 for serial monitor
- Serial.println (“ Arduino Program for input for sensor device using I2C protocol”
- Serial.println (“ Check that Arduino pin SDA bits at pin A4 and serial clock
- SCL pin A5. ”);
- pinMode (internalLED, OUTPUT);
- digitalWrite (internalLED, HIGH);
- //Set I2C serial standard clock rate 100 kHz.
- // Write data at I2DEVICE
- Wire.begin ();
- /* Get the eleven calibration Coefficients from the IC*/
- I2CDeviceCalibration ( ); //Read integers for 11 calibration coefficients
- /* Declare initial offset, for example, Sensor value when temperature is 0
- degree Celsius or pressure is atmospheric pressure or photo-current when dark.
- */
- observedValue0 = 0; // Declare the initial value of the parameter
- /* Calculate sensed Offset value result using calibCoeffs. */

Setup ( )

- Declare
- int observedValue0, observedCValue0,
- pinMode (internalLED, OUTPUT);
- digitalWrite (internalLED, HIGH);
- //Set I2C serial standard clock rate 100 kHz.
- // Write data at I2DEVICE
- Wire.begin ();
- /* Get the eleven calibration Coefficients from the IC*/

Setup ( )

- I2CDeviceCalibration ( ); //Read integers for 11 calibration coefficients
- observedValue0 = 0; // Declare the initial value of the parameter
- /* Calculate sensed Offset value result using calibCoeffs. */
- observedCValue0 = DeviceSensedResult (observedValue0);

loop ( )

- observedValue1 = I2CSerialReadInt (I2DEVICE_DATA1_ADDRESS);
- observedValue2 = I2CSerialReadInt (I2DEVICE_DATA2_ADDRESS);
- //find the result for sensed data
- observedCValue1 = **DeviceSensedResult (observedValue1);**
- observedCValue2 = **DeviceSensedResult (observedValue1);**

Summary

We learnt

- Example 9.5 for Programming the serial reading of RFID tag IC
- RFID IC sending 12 bytes, one for the header (address), ten bytes for the sensed data or RFID tag or ID and one byte for the end character

Summary

We learnt

- Example 5.6 for Programming the serial reading using IC sensor communicating using serial I2C bus
- Program control register and read sensed 2 Byte value and 11 coefficients
- Late observed value.

End of Lesson 4 on
Programming Arduino Examples 9.5 and 9.6