Title: Bit Banging EEPROM with Arduino Nano

Deliverable 1

Create a console program to interface with an EEPROM chip. You will be manually

Resources:

- ❑ Bit Banging SPI Tutorial: https://youtu.be/9hMsNOwY5AQ
- ❑ EEPROM Datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/doc5140.pdf
- ❑ Logic Analyzer Install and doc: https://sigrok.org/wiki/Getting_started_with_a_logic_analyzer

Required Hardware:

- ❑ Arduino Nano
- ❑ Protocol Analyzer, 8 Channel 24MHz
- ❑ EEPROM AT93C46
- ❑ Breadboard, jumper wires and headers

Required Software:

- ❑ PulseView (For use with Protocol Analyzer)
- ❑ Arduino IDE

Setup Protocol Analyzer

- ❑ Install Arduino IDE and make sure you can upload to Arduino
- ❑ Install PulseView (includes Zadig Program)
- ❑ Use the Zadig program to install driver (just plug in analyzer and click Install driver)
- ❑ Connect to Device -> fx2lafw (generiv driver for FX2 based LAs)(fx2lafw) -> Scan -> CWave USBee AX
- ❑ Monitor the protocol using the analyzer for a digital output
- ❑ Use the protocol analyzer to debug SPI communication with the EEPROM

Create an interactive console:

- Use the EWEN/READ/WRITE/ERASE instructions of the EEPROM to create an interactive console that allows the user to read and write data to specific addresses.

  Examples:

  - `WRITE 02 r` should write an 'r' character to address 0x02
  - `READ 7A` should read the data at address 0x7A
- Implement the ERAL and WRAL instructions.
- *NOTE: Make sure to use Teraterm to access the serial interface. Do not rely on the built-in Arduino serial monitor as it does not behave like industry standard console interfaces.*
- Control the ORG pin via software so user can switch between x8 and x16 organization
- Add a memory dump feature that outputs the entire memory structure. The memory dump should output in x8 organization only. The output should include hex and ASCII representation of the data similar to below:

```
Addr  Registers                                       ASCII
0x00  6D 79 20 64 61 74 61 2E 2E 2E 2E 2E 2E 2E 2E 2E    "my data........."
0x10  4D 6F 72 65 20 64 61 74 61 20 73 61 76 65 64 2C    "More data saved."
0x20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20    "                "
...
0x60  30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46    "0123456789ABCDEF"
0x70  4D 6F 72 65 20 64 61 74 61 20 73 61 76 65 64 21    "More data saved!"
```

  NOTE: Non-printable characters can be shown as a period (.)

Deliverable 2

Create a motion detector that senses when an object is nearby and increments a counter each time an object is within a specific distance.

Resources:

- LED&KEY Datasheet: https://jetpackacademy.com/wp-content/uploads/2018/06/TM1638_cheat_sheet_download.pdf
- Ultrasonic Sensor Datasheet: https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf

Required Hardware:

- LED&KEY (8 x LED/Digit/Button)

Required Software:

- None

Control LED&KEY Screen:

- The digits will start at 0 and increment every time an object enters within a 3 foot threshold
- Zero out the counter whenever a specific button is pressed.
- Add a button to increase the distance threshold
- Add a button to decrease the distance threshold
- NOTE: Do not use Arduino's built in shiftOut/shiftIn/pulseIn functions. You will be manually shifting using the digitalWrite() and digitalRead() functions.

Deliverable 3

Resources:

- Small Device C Compiler (SDCC): https://www.instructables.com/8051-Programming-Using-Small-Device-C-Compiler-SDC/
- STC89 Datasheet: http://www.stcmcudata.com/datasheet/stc/STC-AD-PDF/STC89C51RC-english.pdf
- STC ISP Programming Software: https://www.stcmicro.com/rjxz.html

STC89 Demo Board:

- Implement the functionality for deliverable 1 and 2 using Small Device C Compiler

DPS Telecom

"Your Partners in Network Alarm Management"

ISO Cover Sheet

bit-banging

Project Outline for Bit-Banging EEPROM, LED&KEY and STC89 Development

	Policy	Document No	Date	Version	4
	Quality Procedure	39910.1339615	01/20/2023		
X	Working Instruction			Pages w/o Cover	4
	MGR Approval:	R_M - 09/22/2023	**ISO Approval:**	- 09/22/2023	

Version	Date	Employee	Revision Summary
4	01/20/2023	RCH	Made clear the code should not use built in shiftOut, shiftIn, pulseIn function calls.
3	01/17/2023	RCH	Updated STC datasheet, added WRITE/READ examples and now require ERAL/WRAL instructions.
2	01/05/2023	RCH	Initial Upload
1	01/05/2023	RCH	Initializing Document