# Embedded Systems Laboratory Project

Equipment Required:

1 × MSP-EXP430FR5739 Experimenter Board Curtin's Kit (USB programming cable, breadboard, wires, resistors, servo motor:
1 × Arduino
1 × SpikenzieLabs LCD interface with display module (MPTHv3) or HD44780-compatible LCD module with I2C PCF8574T/PCF8574AT interface
1 × Bench set of oscilloscope (i.e. Agilent MSO-X 2012A digital oscilloscope and probes) and multimeter (i.e. Fluke 175 and probes)
1 × PC with Energia software installed
Note: Other institutions may use substitute parts

## 1. Objectives

- Design of a simple embedded system
- Working with sensors and LCDs
- Working is I2C and SPI serial buses
- Communication between two different microcontrollers

## 2. General notes

- You can do the tasks in any order that you want.
- After finishing each task, show the results to your lab supervisor to get marks. You need to explain and justify your code.
- If you are using additional libraries (for example the NTC header files) you need to put the ".ino" files in the same folder as your ".ino" name. Then, put the library files in the same folder. This way, when opening the ".ino" file, the libraries will open automatically.

## 3. Tasks Part 1

1. Connect the MSP430 to your lab computer and write a new program that writes your team members names on the Serial monitor

2. Connect a servo motor to the MSP430. Power the servo motor using an external power supply. Make sure to have common GND between the MSP430, Servo, and the power supply. The aim is to display the temperature on the serial and when the temperature is above a certain level, make the servo do a swiping motion while the temperature is above the threshold (you can set the threshold yourself, depending on the lab temperature on the day of your lab session).

3. Write a program that gets input from the serial monitor to determine a position for the servo.

4. There is a built-in accelerometer sensor on MSP430. Write a program that displays the data on the Serial monitor. You also need to map one of the axis to the LEDs on the board. The aim is to map the LEDs in such a way that moving the MSP430 board changes the LEDs that are turning on. For example, If MSP430 is tilted to on one side, LEDs 4,3,2,1 turn on, one after another and if MSP430 tilts to the opposite side, LEDs 4,5,6,7 turn on, one after another. You can use raw accelerometer data and make sure implement a calibration function using one of the push buttons. This means the calibration button should reset all axis to zero given any position for the board, for example if someone wanted the initial starting position to be tilted to the left, they should put the board tilted to the left, press the calibration button, and make this the new baseline.

5. Combine all the previous tasks into a single program. You need to implement the tasks mentioned above in individual modes and add the ability to switch between them using the built-in push buttons on the board. You have to use the serial monitor to display instruction for what mode you are currently in.

# 4. Tasks Part 2

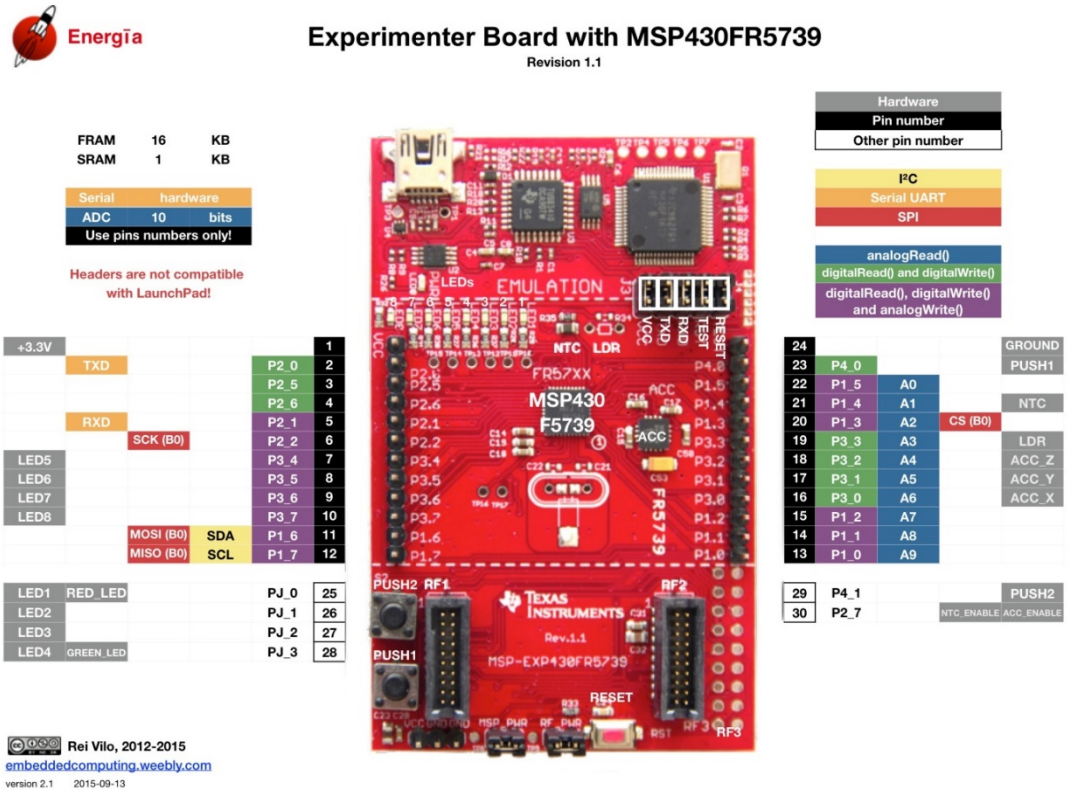
1. Connect the LCD to an Arduino using I2C bus (You can use your own Arduino if you have one).

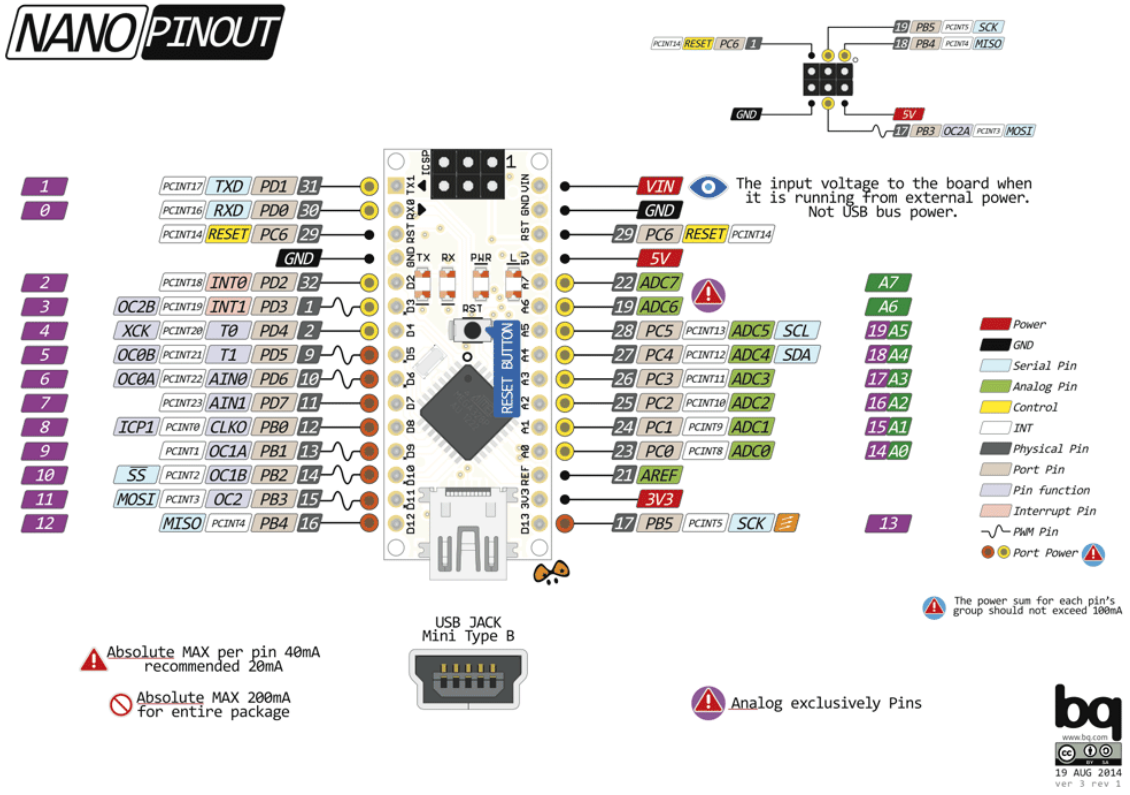
2. Connect the MSP430 to the Arduino using the SPI bus. You are looking for SCLK, MOSI, MISO, and SS connections. The following link has an example of connecting two Arduino boards together witch can be helpful Arduino to Arduino using SPI. You can also use the examples provided on blackboard. Keep in mind to have common ground for everything.

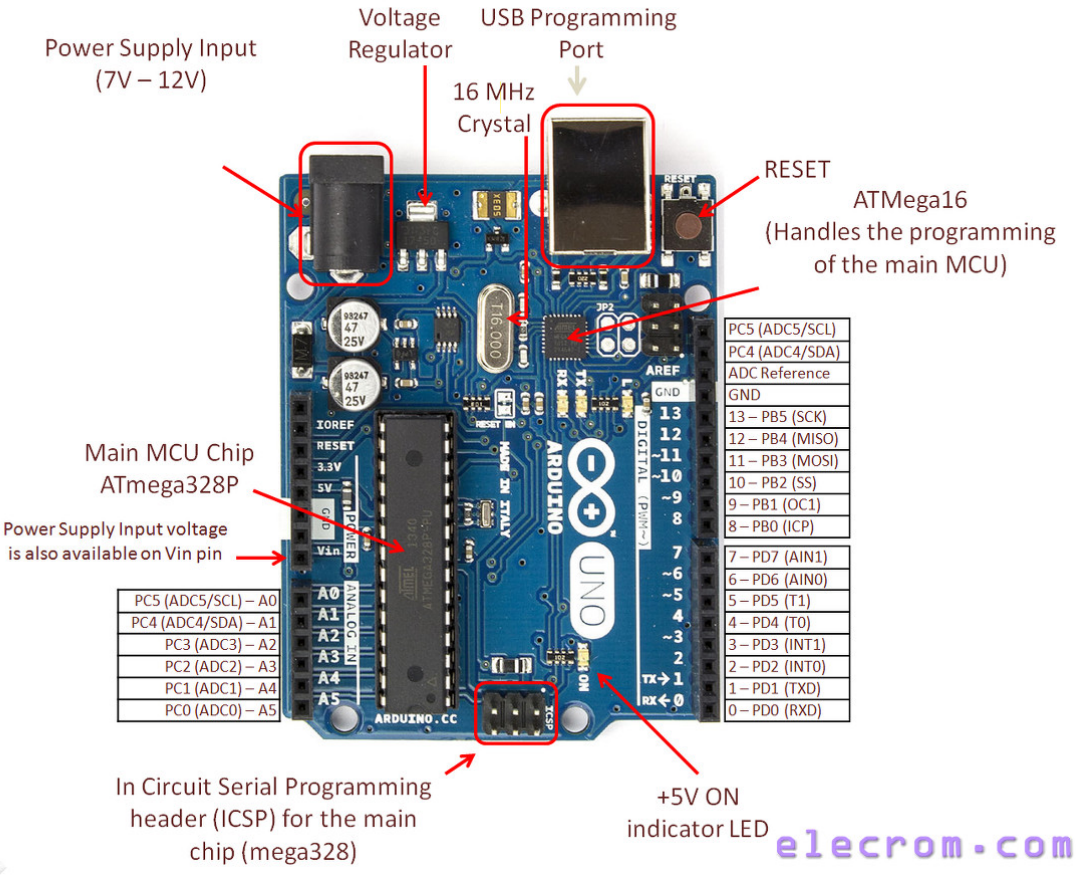
3. The main task is to send data from MSP430 to Arduino and from Arduino to the LCD. You need to do this using the following methods:

    a. Have a string (like hello world) on the MSP430, send it to the Arduino and LCD

    b. Have the string be on the Arduino side and send an initiation signal from the MSP430 so that the Arduino starts writing to the LCD


Energia
Experimenter Board with MSP430FR5739
Revision 1.1
FRAM 16 KB
SRAM 1 KB
Serial hardware
ADC 10 bits
Use pins numbers only!
Headers are not compatible with LaunchPad!
Hardware
Pin number
Other pin number
I²C
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()
+3.3V
TXD
RXD
SCK (B0)
MOSI (B0)
MISO (B0)
SDA
SCL
LED1
LED2
LED3
LED4
RED_LED
GREEN_LED
LED5
LED6
LED7
LED8
P2_0
P2_5
P2_6
P2_1
P2_2
P3_4
P3_5
P3_6
P3_7
P1_6
P1_7
PJ_0
PJ_1
PJ_2
PJ_3
EMULATION
MSP430 F5739
FR57XX
NTC LDR
ACC
PUSH1
PUSH2
RF1
RF2
RF3
TEXAS INSTRUMENTS
MSP-EXP430FR5739
RESET
GROUND
PUSH1
NTC
LDR
ACC_Z
ACC_Y
ACC_X
CS (B0)
NTC_ENABLE
ACC_ENABLE
Rei Vilo, 2012-2015
embeddedcomputing.weebly.com
version 2.1 2015-09-13


NANO PINOUT
The input voltage to the board when it is running from external power. Not USB bus power.
RESET BUTTON
USB JACK Mini Type B
Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package
Analog exclusively Pins
The power sum for each pin's group should not exceed 100mA
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power
19 AUG 2014


Power Supply Input (7V – 12V)
Voltage Regulator
USB Programming Port
16 MHz Crystal
RESET
ATMega16 (Handles the programming of the main MCU)
Main MCU Chip ATmega328P
Power Supply Input voltage is also available on Vin pin
In Circuit Serial Programming header (ICSP) for the main chip (mega328)
+5V ON indicator LED
PC5 (ADC5/SCL) – A0
PC4 (ADC4/SDA) – A1
PC3 (ADC3) – A2
PC2 (ADC2) – A3
PC1 (ADC1) – A4
PC0 (ADC0) – A5
PC5 (ADC5/SCL)
PC4 (ADC4/SDA)
ADC Reference
GND
13 – PB5 (SCK)
12 – PB4 (MISO)
11 – PB3 (MOSI)
10 – PB2 (SS)
9 – PB1 (OC1)
8 – PB0 (ICP)
7 – PD7 (AIN1)
6 – PD6 (AIN0)
5 – PD5 (T1)
4 – PD4 (T0)
3 – PD3 (INT1)
2 – PD2 (INT0)
1 – PD1 (TXD)
0 – PD0 (RXD)
ARDUINO.CC
ARDUINO
UNO
elecrom.com

# Marking Sheet for Laboratory Project

Date and Session Time: _____

Names and ID Numbers: _____

**(Please do not forget to write your student ID number)**

| Task | Assessment Criteria | | | Allocated Mark |
|---|---|---|---|---|
| | **0%–35%** | **35%–70%** | **70%–100%** | |
| Writing the name | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /10 |
| Servo, Motor and Temp | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /15 |
| Accelerometer and LED | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /15 |
| Servo motor input through serial monitor | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /15 |
| Combining tasks from part 1 | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /30 |
| MSP430 and Arduino SPI connection | Students did not attempt the task | Students attempted the task, but it was not fully functional | All tasks completed, and questions answered | /15 |