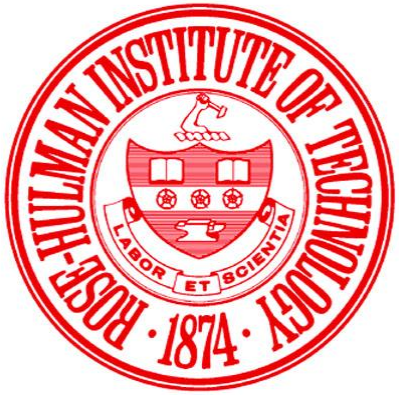


Instrumentation and Microcontrollers using Automatic Code Generation

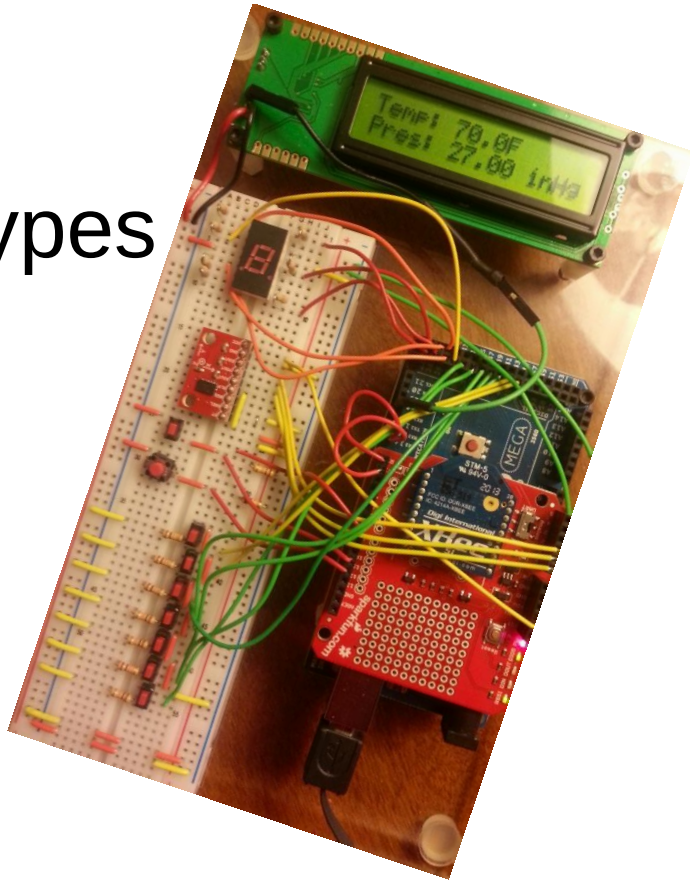
Marc E. Herniter
Professor, Rose-Hulman Institute of Technology

Class Purpose

- Redesign the classic first course in microcontrollers to use Simulink, and automatic code generation to program microcontrollers.
- Open the world of microcontrollers to all fields of study.
  - Open to students of all technical disciplines
  - Target non-Computer Engineering students
- Teach students the basic skills for using microcontrollers and also give them some practical applications like data collection, reading sensors, and generating physical outputs.

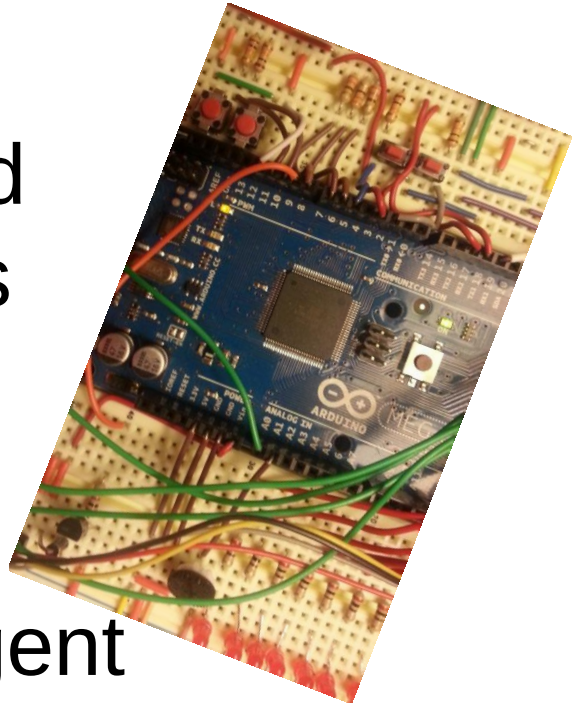
General Topics

- Electronics (Useful Circuits)
- Sensors
- Logic
- Number Systems and Data Types
- ADC and DAC
- Communication Ports
- Programming Methods



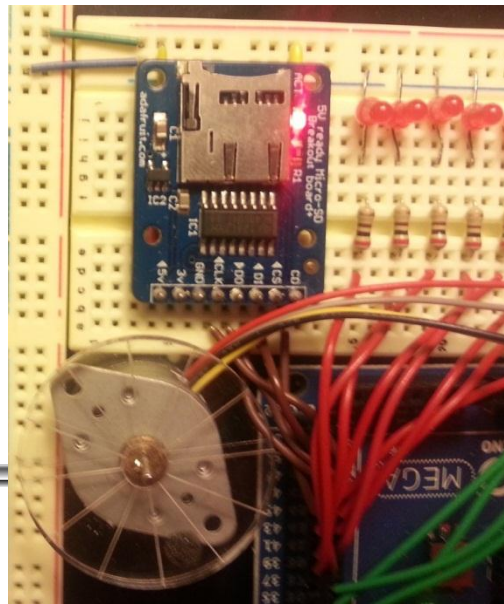
List of Laboratories

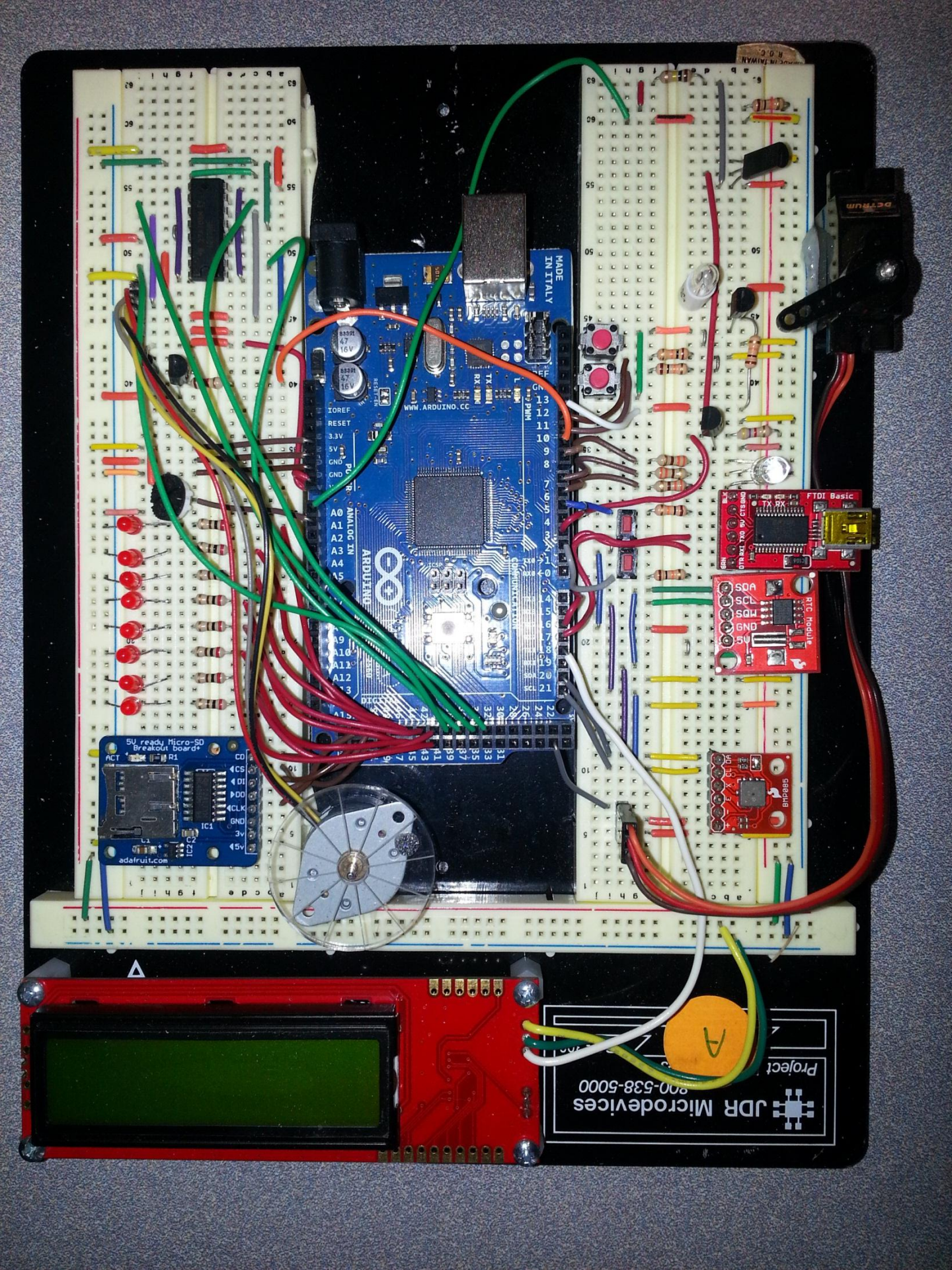
- Hello World!
- Counters, Digital Input, Stateflow, Truth Tables
- Analog Input, Sensors, Triggered Subsystems, and Lookup Tables
- Analog Output (PWM)
- Serial Communication
- S-Functions, Drivers, and Intelligent Sensors (I2C)



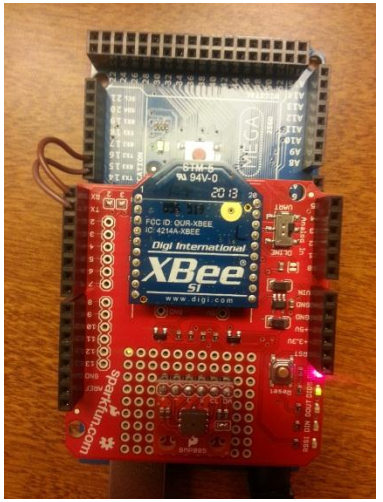
List of Laboratories

- Memory structures and memory types
  - MATLAB structures (RAM)
  - EEPROM (s-function)
- Collecting data using serial communication with MATLAB.
- Collecting data using an SD card (SPI bus)
- Motors
  - Stepper
  - DC Motor
  - Servo Motor




MADE IN ITALY
WWW.ARDUINO.CC
ARDUINO
FTDI Basic
RTC Module
SDA
SCL
SQW
GND
5V
5V ready Micro-SD Breakout board+
adafruit.com
JDR Microdevices
800-538-5000

Course Format

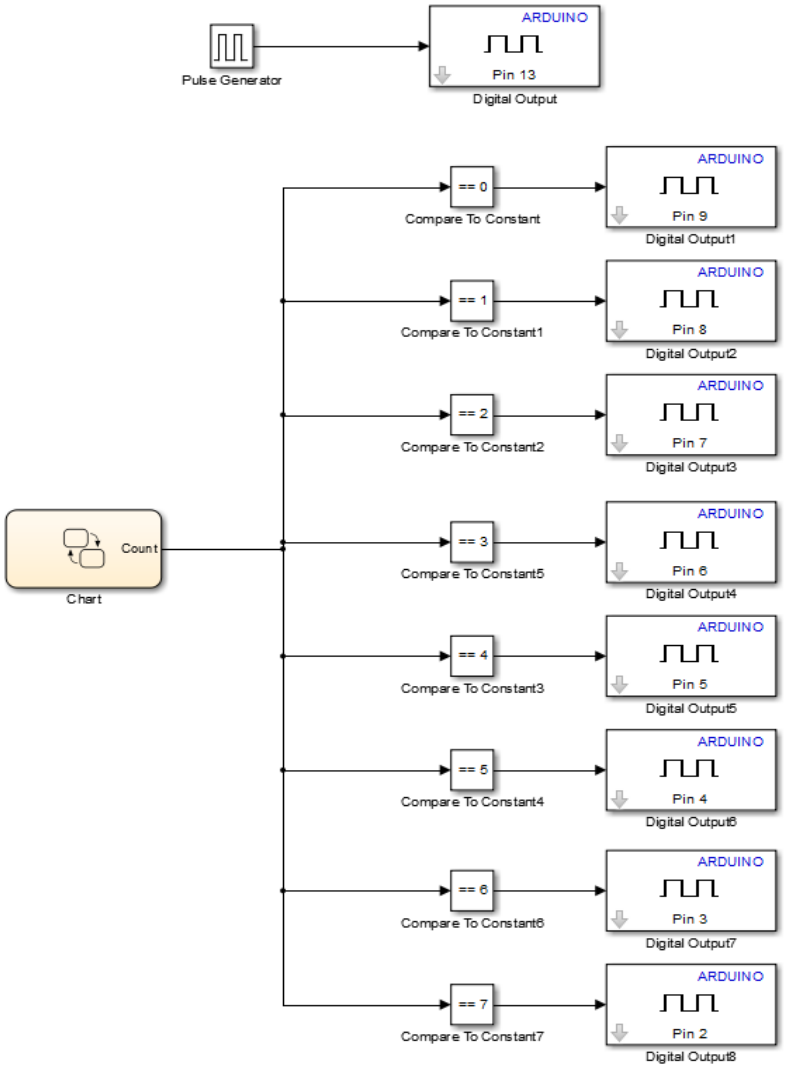


- Two Credit Laboratory Course
- Elective
- Three 2-hour labs
- Hands-on Laboratory
- Students required to demonstrate all programs.
- No exams
- Lot's of work
- All work done on breadboard.
- Use breakout boards for the external circuits:
  - Temp sensor, RTC, SD card reader, accelerometer.

Parts List

- Arduino Mega 2560 R3 ($60)
- DC Motor, Servo Motor, Stepper Motor
- Serial port ($15)
- Serial Enabled LCD Display ($25)
- BMP085 Pressure/Temp Sensor (I2C) ($10)
- TC74 Temperature Sensor (I2C) ($1.50)
- RTC Clock Module (I2C) ($15)
- MicroSD Card reader and SD Card (SPI) ($15)
- Lamps (LED, incandescent), wire kit, breadboard
- Capacitors, thermistor, BJT
- Pushbuttons, potentiometers, resistors
- Stepper motor driver IC (L293)

Counters



- Ring Counter
- Up/Down Counter
- Variable Speed Counter
- Voltmeter
- Logic
  - Simulink Counter
  - Stateflow Counter
  - Memory Block Counter
  - Basic Logic Blocks
  - Truth Table Decoder

Truth Tables

Block: LAB2_Project6/Truth Table

File Edit Settings Add Help

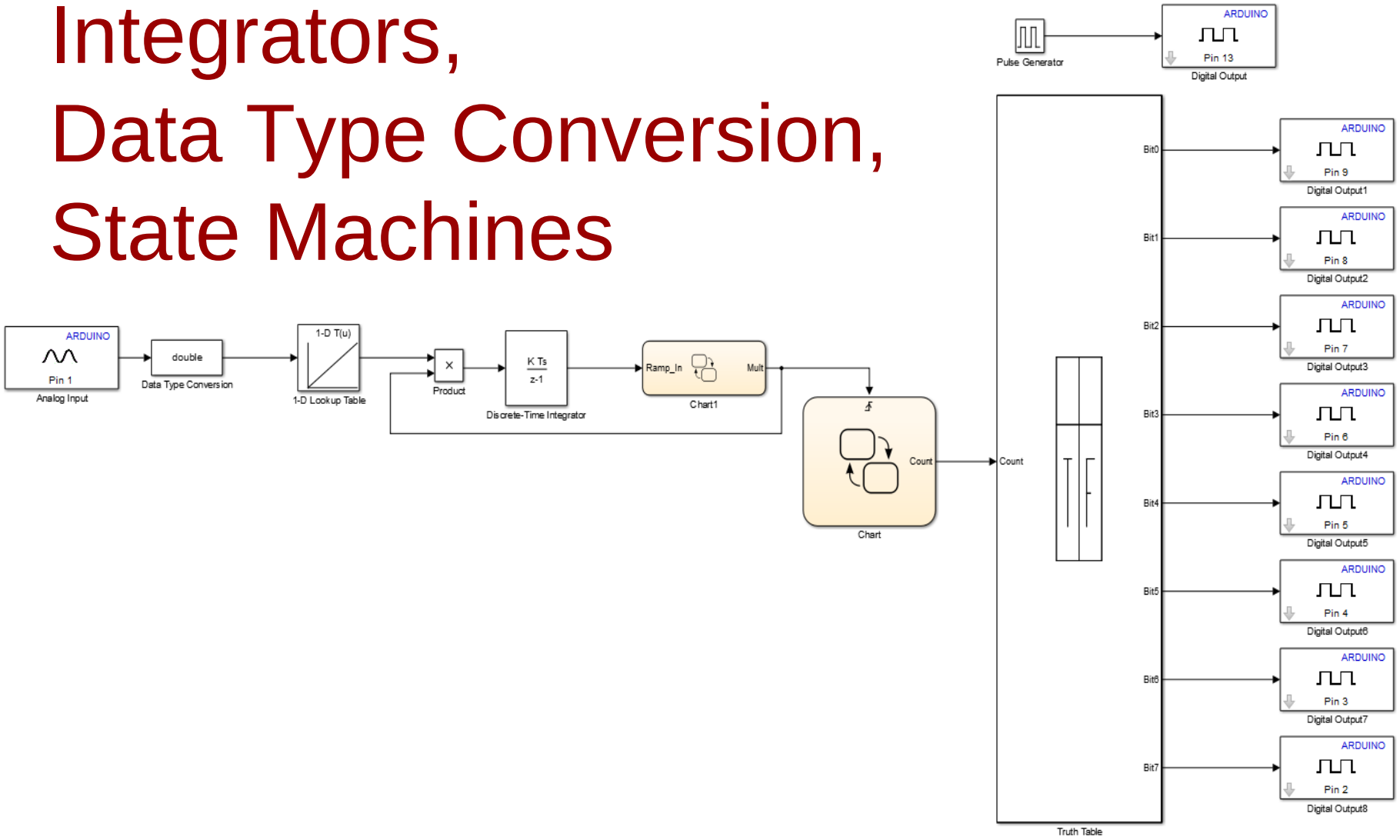
Condition Table

	Description	Condition	D1	D2	D3	D4	D5	D6	D7	D8	D9
1	Count is zero.	Count==0	T	F	F	F	F	F	F	F	–
2	Count is one.	Count==1	F	T	F	F	F	F	F	F	–
3	Count is two.	Count==2	F	F	T	F	F	F	F	F	–
4	Count is three.	Count==3	F	F	F	T	F	F	F	F	–
5	Count is four.	Count==4	F	F	F	F	T	F	F	F	–
6	Count is five.	Count==5	F	F	F	F	F	T	F	F	–
7	Count is six.	Count==6	F	F	F	F	F	F	T	F	–
8	Count is seven.	Count==7	F	F	F	F	F	F	F	T	–
		Actions: Specify a row from the Action Table	1	2	3	4	5	6	7	8	1

Action Table

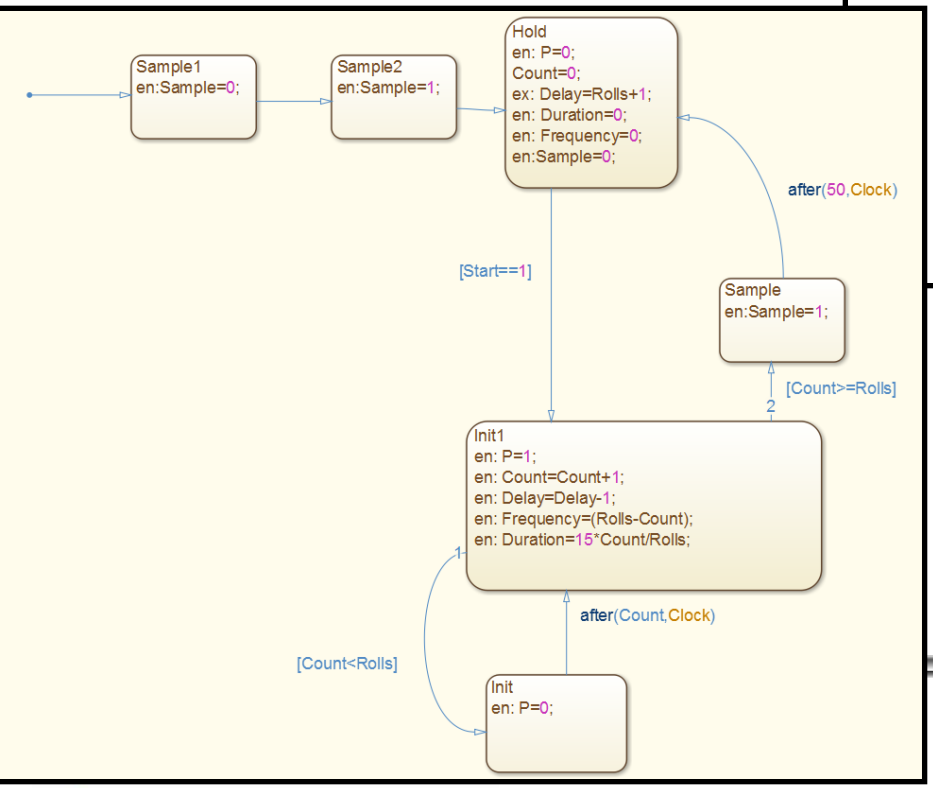
#	Description	Action
1	Light up bit 0.	Bit0=1;Bit1=0;Bit2=0;Bit3=0;Bit4=0;Bit5=0;Bit6=0;Bit7=0;
2	Light up Bit 1.	Bit0=0;Bit1=1;Bit2=0;Bit3=0;Bit4=0;Bit5=0;Bit6=0;Bit7=0;
3	Light up Bit 2.	Bit0=0;Bit1=0;Bit2=1;Bit3=0;Bit4=0;Bit5=0;Bit6=0;Bit7=0;
4	Light up Bit 3.	Bit0=0;Bit1=0;Bit2=0;Bit3=1;Bit4=0;Bit5=0;Bit6=0;Bit7=0;
5	Light up Bit 4.	Bit0=0;Bit1=0;Bit2=0;Bit3=0;Bit4=1;Bit5=0;Bit6=0;Bit7=0;
6	Light up Bit 5.	Bit0=0;Bit1=0;Bit2=0;Bit3=0;Bit4=0;Bit5=1;Bit6=0;Bit7=0;
7	Light up Bit 6.	Bit0=0;Bit1=0;Bit2=0;Bit3=0;Bit4=0;Bit5=0;Bit6=1;Bit7=0;
8	Light up Bit 7.	Bit0=0;Bit1=0;Bit2=0;Bit3=0;Bit4=0;Bit5=0;Bit6=0;Bit7=1;

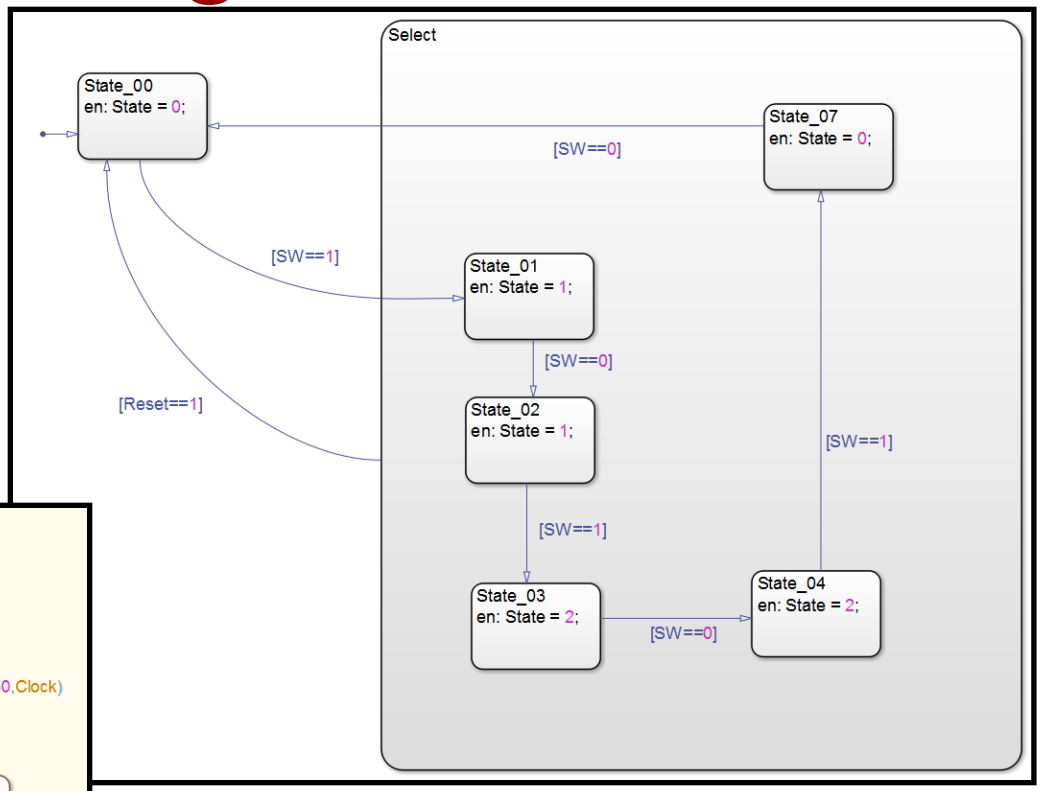
Lookup Tables
Integrators,
Data Type Conversion,
State Machines



State Machine Logic

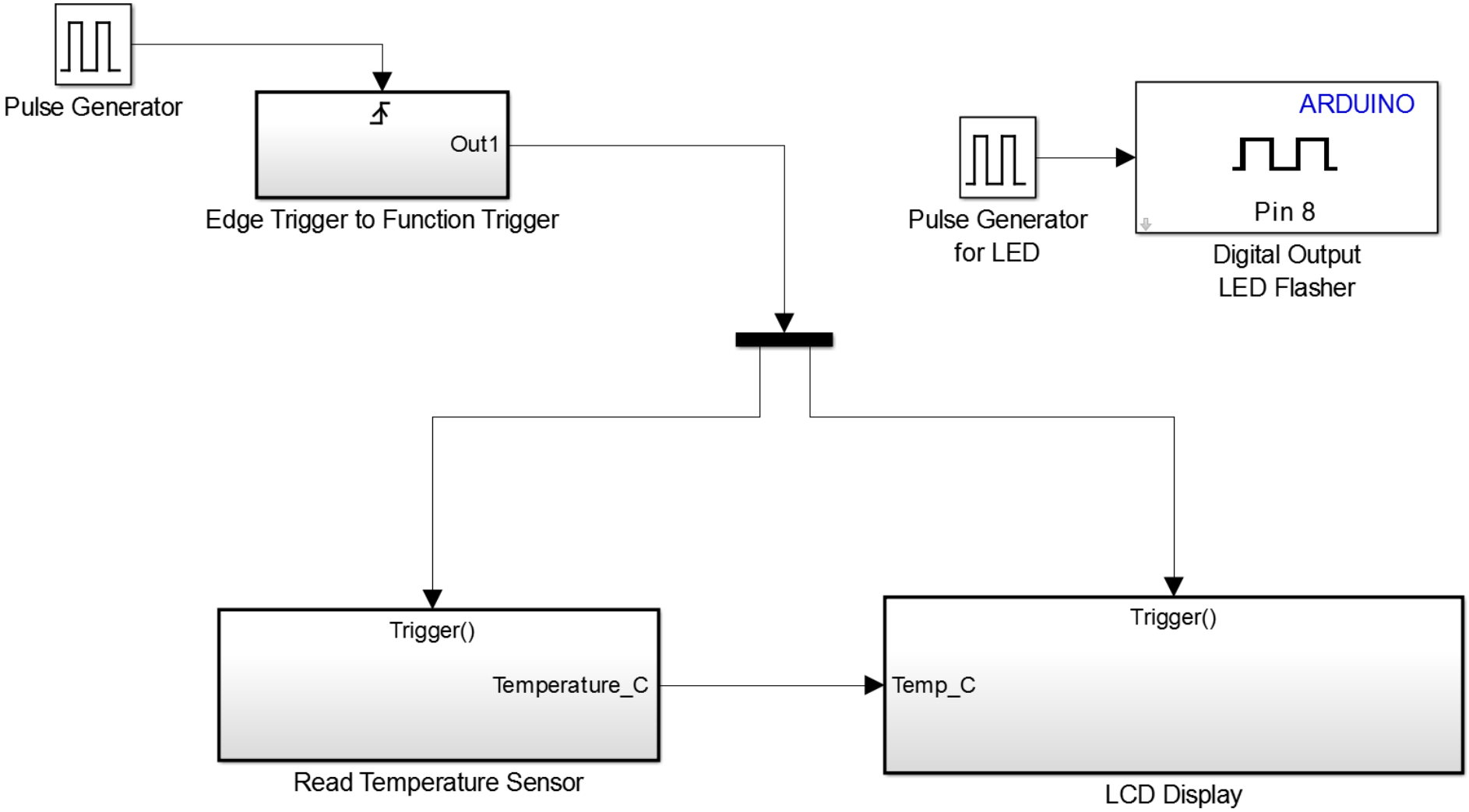
Die Roll Counter / Tone Frequency Calculator



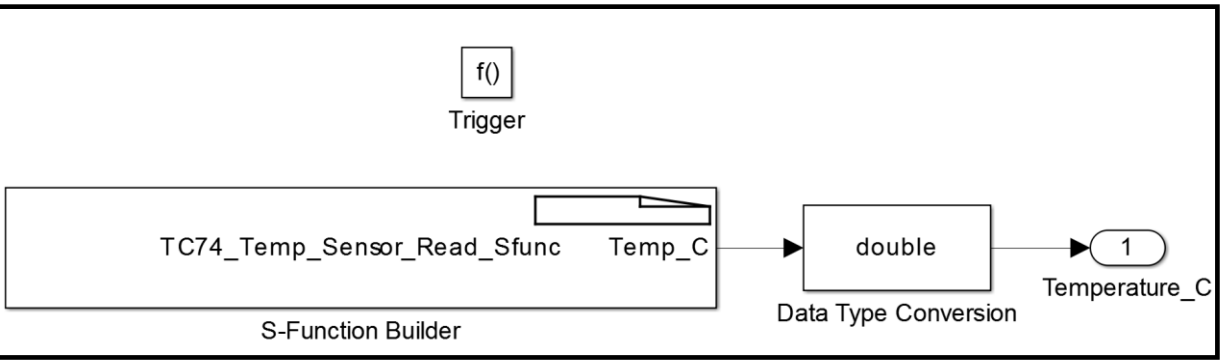


Pushbutton Selector Switch

Triggered Subsystems



I2C Temperature Sensor



S-Function Builder: Lab6_Model1/Read Temperature Sensor/S-Function Builder

Parameters

S-function name: TC74_Temp_Sensor_Read_Sfunc

S-function parameters

Name	Data type

Port/Parameter

- Input Ports
- Output Ports
  - Temp_C
- Parameters

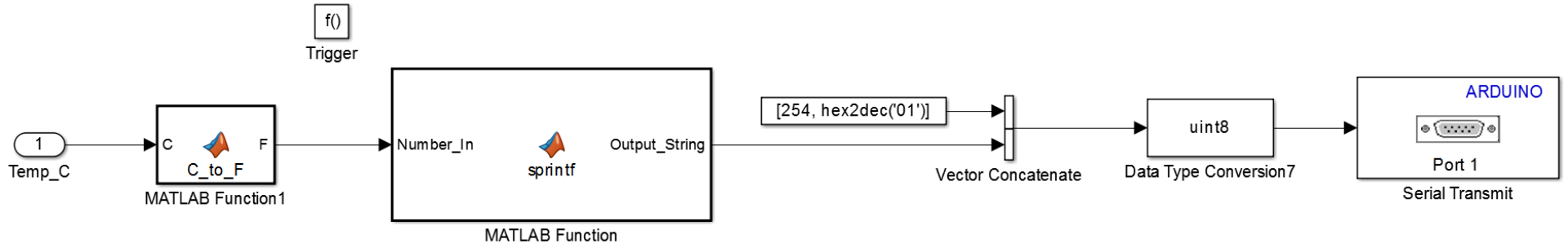
Initialization | Data Properties | Libraries | Outputs | Continuous Derivatives | Discrete Update | Build Info

Code description

Enter your C-code or call your algorithm. If available, discrete and continuous states should be referenced as, xD[0]...xD[n], xC[0]...xC[n] respectively. Input ports, output ports and parameters should be referenced using the symbols specified in the Data Properties. These references appear directly in the generated S-function.

```
/* wait until after initialization is done */
if (xD[0]!=0) {

    /* don't do anything for mex file generation */
    # ifndef MATLAB_MEX_FILE

    Temp_C[0]=sensor.read();

    # endif

}
```

MATLAB Functions



Block: Lab6_Model1/LCD Display/MATLAB Function

EDITOR VIEW

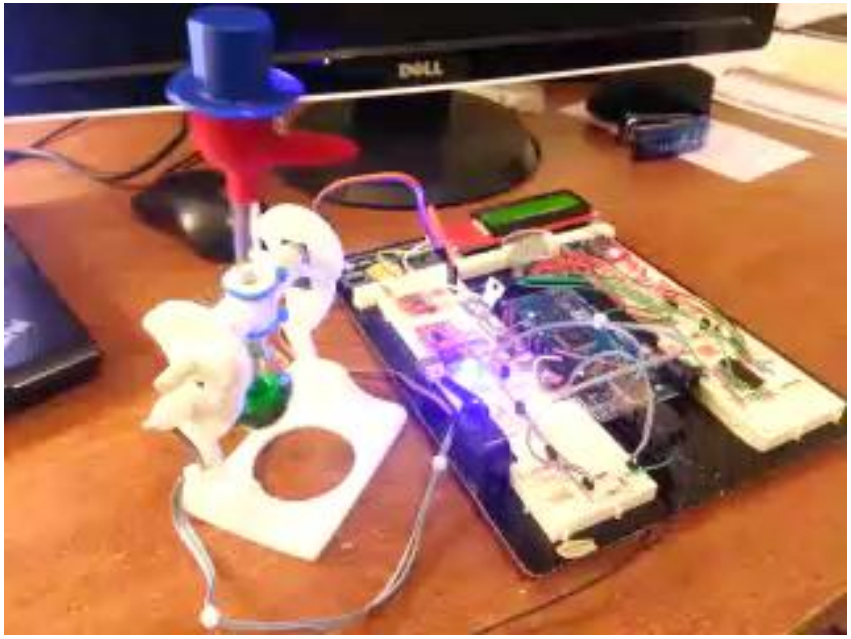
New Open Save Find Files Compare Print | Insert Comment Indent | Go To Find | Breakpoints | Run Model Stop Model Build Model | Go To Diagram Edit Data Simulation Target View Report Help

FILE EDIT NAVIGATE BREAKPOINTS RUN SIMULINK

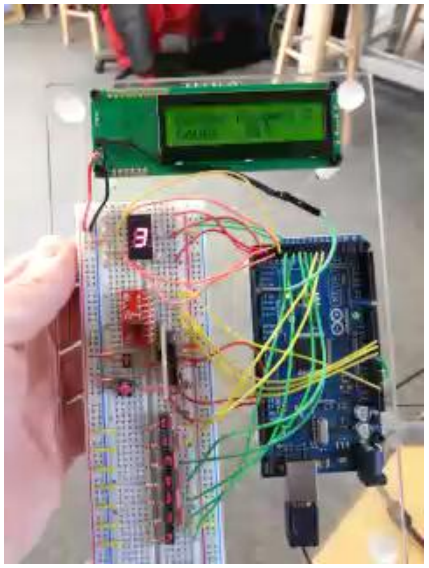
MBSD0 Lab X Nano Android NTX Piano

```
1  function Output_String  = sprintf(Number_In)
2  % This block supports an embeddable subset of the MATLAB language.
3  % See the help menu for details.
4
5  %% Allocate Storage
6
7  lenMax = 32; % Maximum string length including null padding.
8  data_array = ones(1,lenMax, 'uint8').*32;
9  Output_String = ones(lenMax,1, 'uint8').*32;
10
11 %%
12 coder.ceval('sprintf',coder.wref(data_array),coder.opaque('const char *','"The temperature is: %4.1f F.     "'),Number_In);
13
14 data_array(data_array==0)=32;
15 Output_String=data_array(1:lenMax)';
16
17 end
```

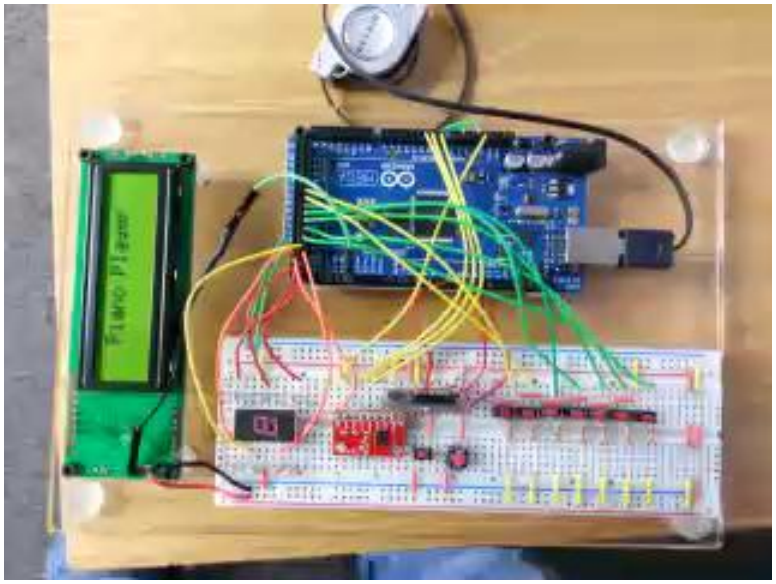
Drinking Bird Controller



Random Dice

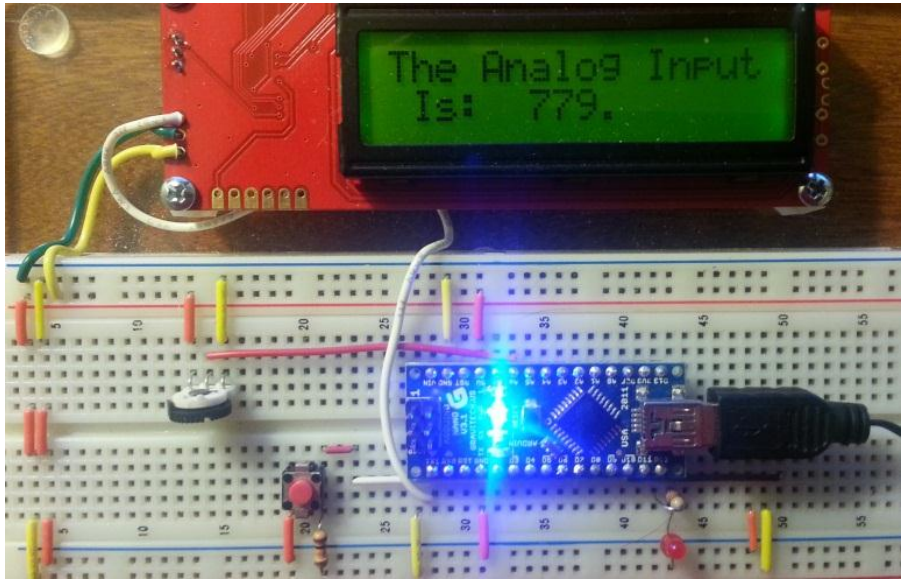


Music Player



Software Requirements

- MATLAB and Simulink Student Suite
  - MATLAB
  - Simulink
  - Control System Toolbox
  - Signal Processing Toolbox
- Additional tools
  - Stateflow
  - Arduino Support Package

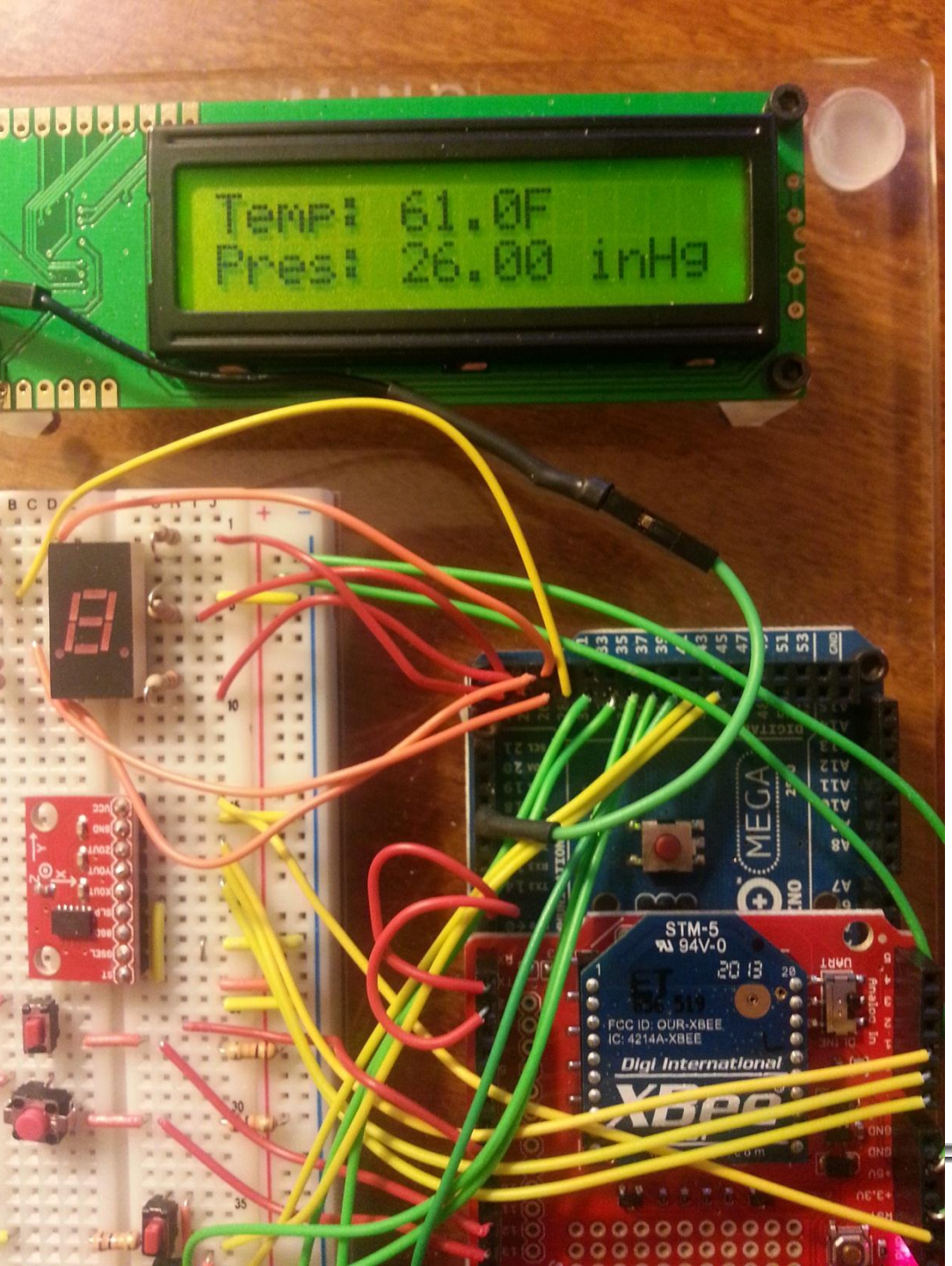


Lessons Learned

- Non-computer engineers can get a quick start easily.
  - Tool chain is simple
  - Communication with target is simple
- Using Simulink to program an Arduino has a much easier learning curve than using C.
  - Programming basic functions, complex look-up tables, and state machines are greatly simplified.
- Students can quickly get to the point where they can implement non-trivial algorithms on a microcontroller with a high degree of functionality.
  - Non-computer engineers would have difficulty getting to this point if they were using C.

Contact Information

- Marc E. Herniter
  - (812) 249-0159
  - Marc.Herniter@ieee.org
  - http://wiki.ece.rose-hulman.edu/herniter
- Course Information
  - http://wiki.ece.rose-hulman.edu/herniter/index.php/Instrumentation_and_Microcontrollers_using_Automatic_Code_Generation
  - Lab Manual, Downloads, Datasheets


Temp: 61.0F
Pres: 26.00 inHg
STM-5
94V-0
FCC ID: OUR-XBEE
IC: 4214A-XBEE
Digi International

Questions?