North Texas NTMS Microwave Society

Arduino Microcontrollers

Greg McIntire, AA5C

AA5C@arrl.net

April 4, 2020

Microcontrollers

- Microprocessors first appeared in the late 1970s
  - Integrated an entire central processing unit (CPU) onto one chip
  - I used an Intel 8051 as a RTTY code/speed converter in the early 1980s (see QST 1982)
- Microcontrollers evolved from microprocessors when memory and other peripheral functions were integrated onto the chip
  - RAM, ROM, EEPROM
  - ADC/DACs
  - ….
- You will find a microcontroller in most modern appliances
- Modern phase locked loop (PLL) ICs require a microcontroller for control
- Some PLL boards include a embedded, pre-programmed microcontroller
  - e.g., ZL boards, DigiLO, Orion (Orion based on 8051!)
- Others PLL boards require an external microcontroller
  - e.g., Chinese ADF4351 development boards

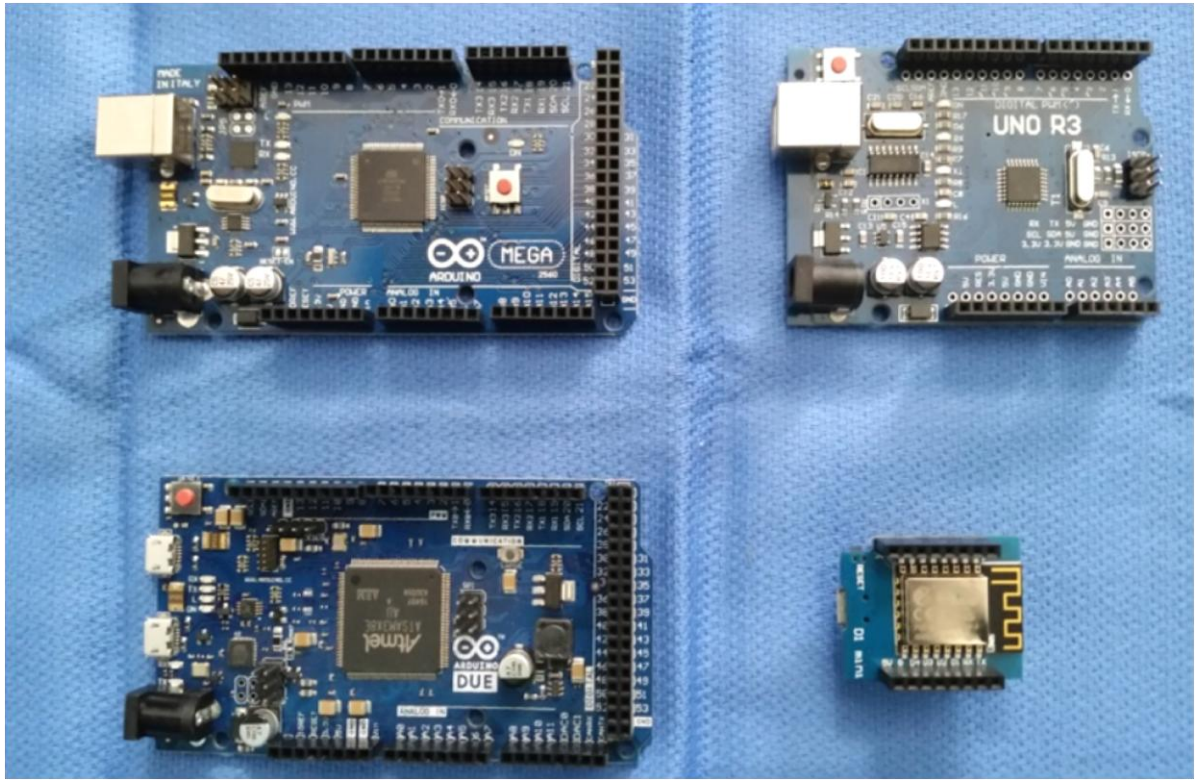
Why “User” Programmable Microcontrollers?

- That is; Arduinos, Raspberry Pi, WeMos, etc.
- Highly flexible and can be programmed for a wide variety of functions
  - Bus control
    - Serial Peripheral Interface (SPI), I2C bus, discrete lines
  - Morse code generator/beacon
  - Relay control
  - Position indication.....
- Inexpensive
- Wide range of supporting hardware extensions available
  - Displays
  - Switches
  - Relays……
- Very large user base with an abundance of open source software

North Texas NTMS Microwave Society

Example Microcontrollers

Arduino Mega 2560



Arduino Uno R3

Arduino Due

WeMos D1 Mini

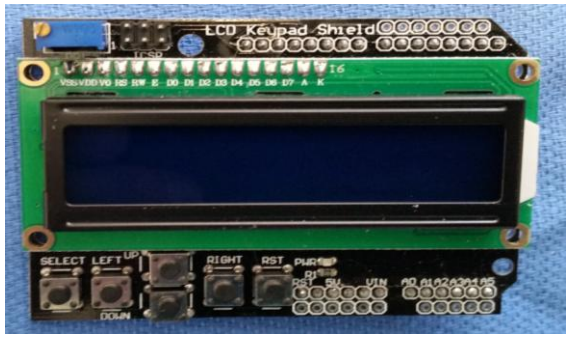
Specifications

North Texas NTMS Microwave Society

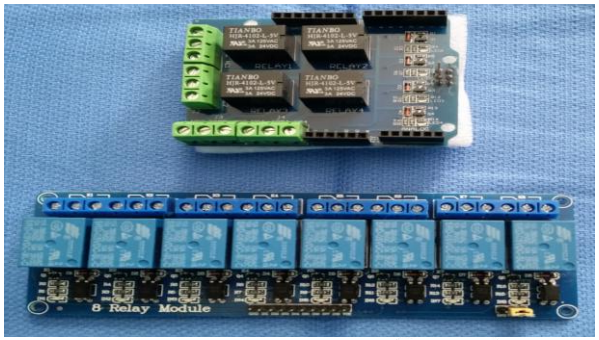
Board	Mega 2560	Uno	Due	D1 Mini
Processor	ATmega2560	ATmega328P	ATSAM3X8E	ESP8266
Clock Speed (MHz)	16	16*	84	80/160
Word Length (bits)	32	8	32	32
I/O Logic Voltage	3.3	5	3.3	3.3
Analog I/O (in/out)	16/0	6/0	12/2	1/0
Digital (I/O or PWM)	54/15	14/6	54/12	11
EEPROM (KB)	4	1	0	512 bytes
SRAM (KB)	8	2	96	4 MB
Flash (KB)	256	32	512	4 MB
SPI Bus Interfaces	1	1	2	1
USB Interface	1-Regular	1-Regular	2-Micro	1-Micro

North Texas NTMS Microwave Society

Shield Examples

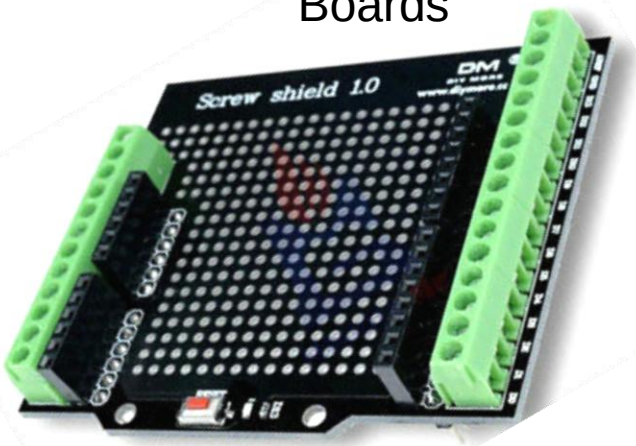


LCD Display With Switches and I/O Breakouts



Four and Eight Relay Boards

Ethernet with SD Card Reader



Prototyping Boards

Programming

- You don’t need to be a software engineer to write code for popular microcontrollers
- A lot of already programmed library functions available, for example
  - SPI bus
  - EEPROM
  - Ethernet
  - LCD display
  - Servo
  - WiFi
  - I2C…….
- Lots of open source software available
  - Often you can start with existing code and modify for your application
- The software task can be mostly one of integration
- On the other hand, the Arduino instruction set is very powerful and you can design very sophisticated programs.

Firmware

North Texas NTMS Microwave Society

- Firmware for Arduino microcontrollers is developed using the Arduino Integrated Development Environment (IDE)
- The Arduino IDE is freeware located at:
  https://www.arduino.cc/en/main/software
- Current version is 1.8.12
- The coding language is very much like C++
- Programs are called sketches
- The IDE facilitates:
  - Writing the sketch
  - Compiling the sketch
  - Uploading the firmware (compiled sketch) to the target Arduino Board
- A USB-A to USB-B (or micro-USB) cable from your computer to the microcontoller is used to upload the firmware

Framework for an Arduino Sketch

North Texas NTMS Microwave Society

1. Comments
   - Anything between /* and */
   - After // for a single line comment
   - Use to document details on how to use the sketch
2. Variables
   - A place for storing a piece of data
3. Functions
   - A procedure or subroutine that can be called from elsewhere in the sketch
4. Setup
   - Called once when the sketch starts
5. Loop
   - Repeatedly executed

Comments & Variable


```
//  ****************************************** HARDWARE IMPORTANT*************************************************
//  With an Arduino UN0 : use a resistive divider to reduce the voltage, MOSI (pin 11) to
//  ADF DATA, SCK (pin13) to ADF CLK, Select (PIN 3) to ADF LE
//  Resistive divider 560 Ohm with 1000 Ohm to ground on Arduino pins 11, 13 et 3 to adapt from 5V
//  to 3.3V the digital signals DATA, CLK and LE send by the Arduino.
//  Arduino pin 2 (for lock detection) directly connected to ADF5355 card MUXOUT.
#include <LiquidCrystal.h>
#include <SPI.h>
#define ADF4351_LE 3
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
byte poscursor = 0; //position cursor common 0 to 15
byte line = 0; // display line in progress on LCD in cours 0 to 1
byte memoire,RWtemp; // number the EEPROM memory
uint32_t registers[6] = {0x00AA0020, 0x08008029, 0x00004E42, 0x000004B3, 0x0085003C, 0x00580005}; //3408 MHz with 10 MHz
reference
int lcd_key = 0;
int adc_key_in  = 0;
int timer = 0,timer2=0; // use to measure the time a key is pressed
unsigned int i = 0;
double RFout, REFin, INT, PFDRFout, OutputChannelSpacing, FRACF;
double RFoutMin = 35, RFoutMax = 4400, REFinMax = 250, PDFMax = 32;
unsigned int long RFint,RFintold,INTA,RFcalc,PDRFout, MOD, FRAC;
byte OutputDivider;byte lock=2;
unsigned int long reg0, reg1;
```

Functions


```
void WriteRegister32(const uint32_t value)   //Program register 32bits
{
  digitalWrite(ADF4351_LE, LOW);
  for (int i = 3; i >= 0; i--)          // loop on 4 x 8bits
  SPI.transfer((value >> 8 * i) & 0xFF); // Offset, hiding the byte and sending via SPI
  digitalWrite(ADF4351_LE, HIGH);
  digitalWrite(ADF4351_LE, LOW);
}

void SetADF4351()  // Program all the registers of the ADF4351
{ for (int i = 5; i >= 0; i--)  // program ADF4351 starting with R5
    WriteRegister32(registers[i]);
}
```

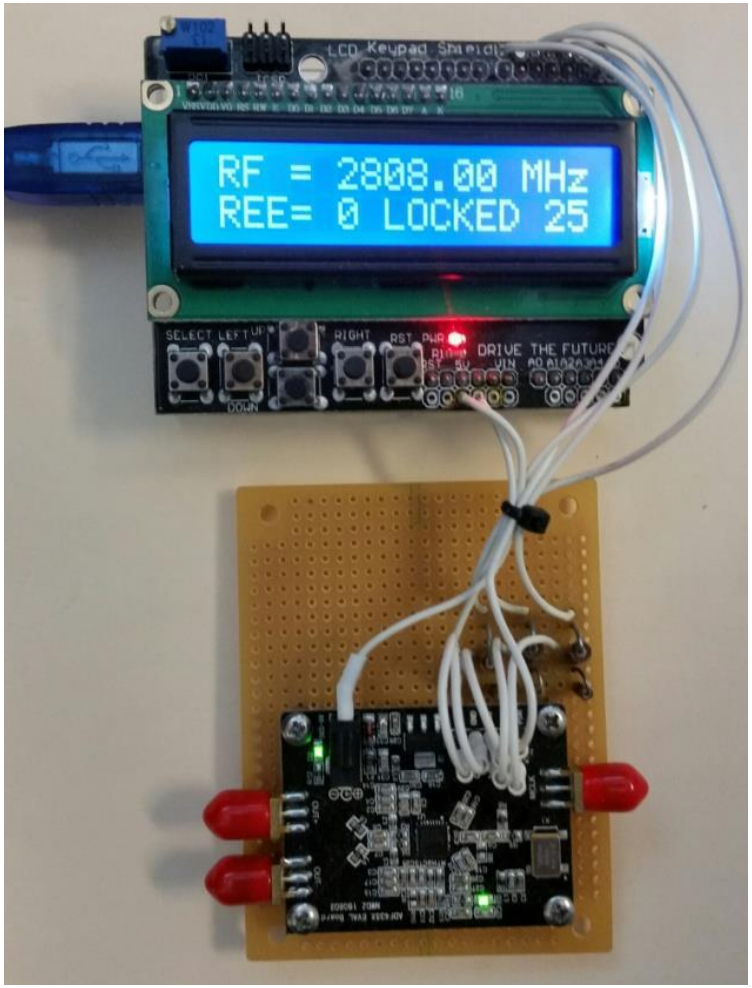
Setup & Loop

North Texas NTMS Microwave Society

```
//********************************** Setup ***************************************
void setup() {
  lcd.begin(16, 2); // two 16 characters lines
  lcd.display();
  analogWrite(10,255); //LCD brightness
  Serial.begin (19200); //  Serial to the PC via Arduino "Serial Monitor"  at 9600
  delay(1000);
  pinMode(2, INPUT);  // PIN 2 in enter to lock
  pinMode(ADF4351_LE, OUTPUT);          // Setup pins
  digitalWrite(ADF4351_LE, HIGH);
  SPI.begin();                          // Init SPI bus
  SPI.setDataMode(SPI_MODE0);           // CPHA = 0 et Clock positive
  SPI.setBitOrder(MSBFIRST);            // poids forts in head
  SetADF4351();  // Program all the registers of the ADF4351
// End setup for ADF4351 code
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.write("AA5C RF=2556 MHz");
  lcd.setCursor(0,1);
  lcd.write("    Ref=25 MHz ");
}
// End setup
//**********************************Loop*********************************
void loop()
 {
  }   // end loop
```

ADF4351 Controller

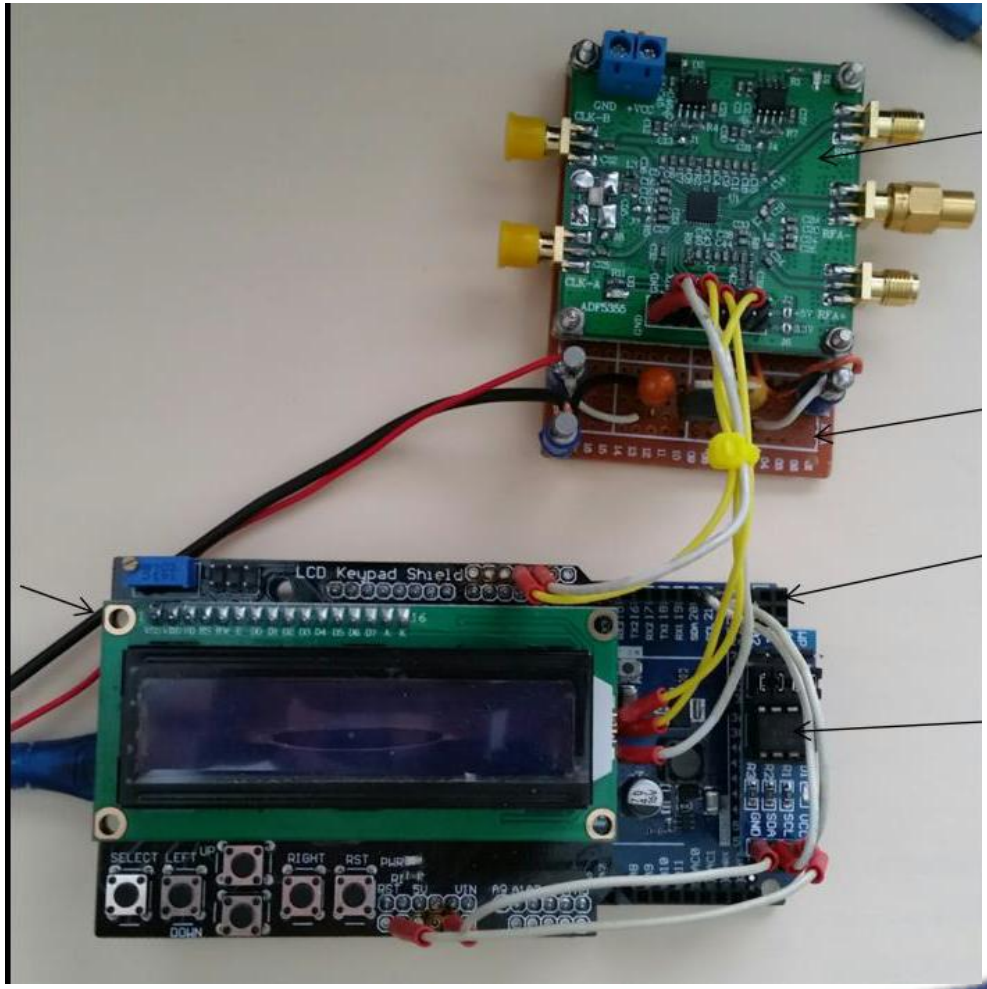
- Analog Devices ADF4351
  - 35 MHz to 4.4 GHz Synthesizer
  - SPI Bus Controlled
  - Programmable features
- Evaluation Board
  - Mounted part
  - 3.3V Regulator
  - On-board reference oscillator
- Interface Board
  - 5V to 3.3V logic conversion
  - Breadboard space
- Arduino Uno R3 Controller
  - SPI bus control
  - Frequency selection
  - Reference selection
  - Display



North Texas NTMS Microwave Society

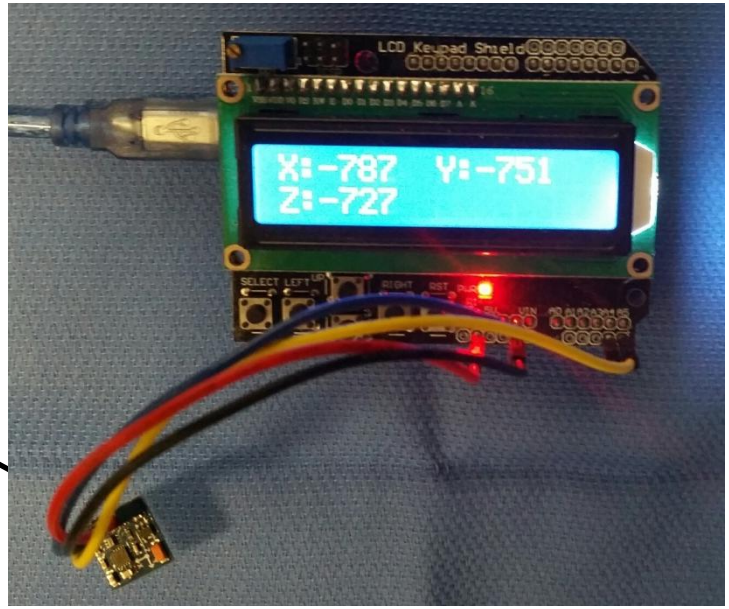
ADF5355 Controller

- Analog Devices ADF5355
  - 54 MHz to 13.4 GHz Synthesizer
  - SPI Bus Controlled
  - Programmable features
- Evaluation Board
  - Mounted part
  - 3.3V and 5V Regulators
  - On-board reference oscillator
- Interface Board
  - Regulator
  - Breadboard space
- Arduino Due Controller
  - SPI bus control
  - Frequency selection
  - Reference selection
  - Display



Position Indication

- HMC5388L 3 Axis Magnetic Compass
  - GY-271 module
- UNIK MPU-9250 9DOF Module
  - Nine-axis Attitude Gyro
  - Compass Acceleration
  - Magnetic Field Sensor
- Both Modules Controlled via an I2C Interface



Calibration in the Target Application Required
- Sensitive to Ferrous Materials

Summary

North Texas NTMS Microwave Society

- Low Cost Microcontrollers Can Be Used for a Wide Range of Functions
  - Controllers
  - Indicators
  - Sensors
- Many Different Supporting Hardware Devices/Shields Available
- Large base of Open Source Software Available
  - Minimizes the Amount of Unique Programming Required
  - Generic Functions, e.g.,
    - SPI bus control, I2C bus control
    - External EEPROM
  - Application Specific Functions, e.g.,
    - Synthesizer Control
    - Rotor Control
    - Az/El Indicators

Next Meeting – NanoVNA V2

North Texas NTMS Microwave Society