Project 3.5 SAR ADC (Successive Approximation Register Analog to Digital Converter)

Purpose

The purpose of the *SAR ADC* (successive approximation register analog to digital converter) is both to demonstrate how voltages are read by computers, and to map an unknown analog voltage to a 7-bit digital number using the successive approximation algorithm.

Reference

Project description: http://darcy.rsgc.on.ca/ACES/TEI4M/SARADC/index.html

Project 3.5.1 Overview and Clock

Purpose

The purpose of the clock is to synchronize the operations of the SAR ADC. Since it uses successive approximation, multiple actions must take place, which warrants the use of a clock signal (for sequential logic).

Reference

Project description: http://darcy.rsgc.on.ca/ACES/TEI4M/SARADC/index.html#task

Project schematic: https://crcit.net/c/f705567608d5456ba69e7a610a942e96

https://dewesoft.com/blog/types-of-adc-converters

Theory

When a computer or digital circuit needs to interact with "real world" signals or quantities (which are generally analog), a digital approximation of that signal must be created. To do this, an ADC is utilized. There are two main attributes of an ADC: resolution, and sample rate. Resolution refers to the precision of the approximation. A more precise (higher resolution) ADC generates an approximation closer to the original analog signal (see Figure 1). A greater sample rate correlates with fewer changes between samples, or a smoother curve, as exhibited in Figure 1.

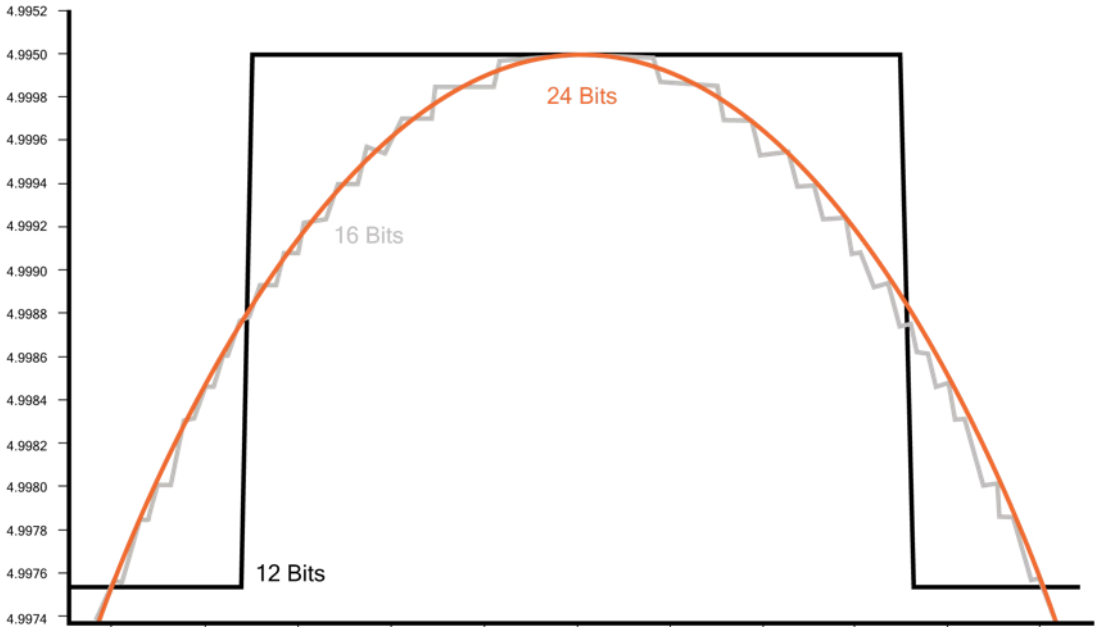


Figure 1. ADC Precision Comparison

Procedure

There are several types of analog to digital converters. One of the most commonly used, however, is the successive approximation register (SAR) analog to digital converter (ADC). This type of ADC approximates an analog signal in several steps, starting with the MSB and working down the LSB.

To generate the approximation, an op amp is utilized as a comparator (see Figure 1). In this configuration, when the voltage presented on the non-inverting input (+ input) is greater than the voltage presented on the inverting input (- input), the output pin is high. When the voltage on the non-inverting input is less than the voltage on the inverting input, the output is low.

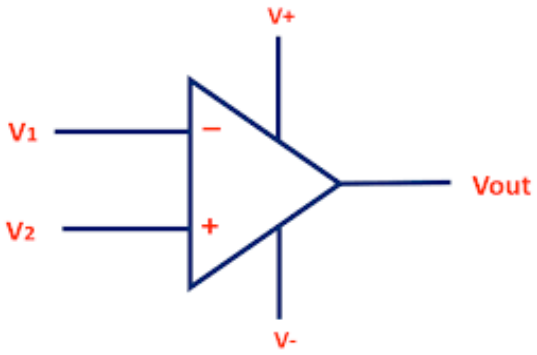


Figure 1. Op Amp in Comparator Configuration

In this configuration, a comparator can be used to determine whether or not an unknown voltage on the non-inverting input is above a threshold, which is set by the reference voltage that is present on the inverting pin (see Figure 1).

Since the sampling process does not happen instantaneously, and the voltage can fluctuate while the sample is being calculated, a *sample and hold* circuit is utilized (see Figure 2). A sample and hold circuit essentially stores the presented voltage for a set amount of time, in the case of the SAR ADC, the sampling duration.

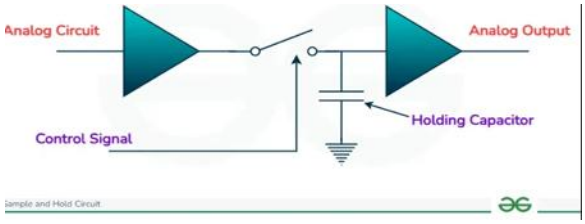
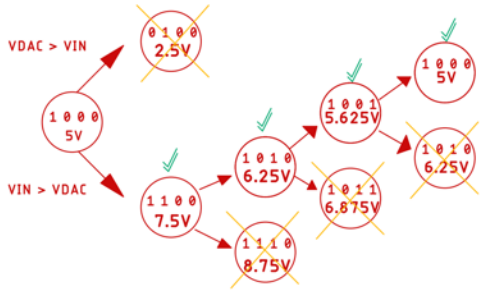


Figure 2. Sample and Hold Circuit

The SAR ADC utilizes a single comparator, with its reference voltage controlled by an R/2R DAC (see Project 2.1 and Project 3.3). On the first iteration, once the voltage is stored in the sample and hold circuit, the DAC has its MSB set and its other bits cleared. In a 5 V system, this sets the reference voltage to 2.5 V. The output of the comparator is fed into the output register, so if the voltage being sampled is greater than 2.5 V, a 1 is placed into the MSB of the output register. If the voltage is less than 2.5 V, a 0 is placed into the MSB of the output register. Additionally, the output register is tied back to the DAC through control logic. In this configuration, the DAC receives the already-calculated bits.



After the MSB has been calculated, the second MSB is calculated by setting the second MSB. This solves the second MSB, in a similar fashion to the MSB. This process is successively repeated, until each bit has been solved for.

Figure 3. Successive Approximation Binary Tree

Figure 3 depicts the successive approximation algorithm using a binary tree. In Figure 3, the sampled voltage is 5 V in a 9 V system.

The binary tree (see Figure 4) is closely related to the successive approximation algorithm, as the SAR algorithm uses a *binary search algorithm*. A binary search algorithm involves starting at the MSB, and having a node for each bit, which allows for two decisions.

By utilizing a binary tree structure, the SAR ADC ensures that the sampling process executes in a predictable number of clock cycles. If the ADC had to cycle through each possible binary value, the total time to sample would vary greatly depending on the sampled voltage. Additionally, it allows the SAR ADC to operate relatively quickly. Some SAR ADCs sample at rates in excess of 10 MHz.

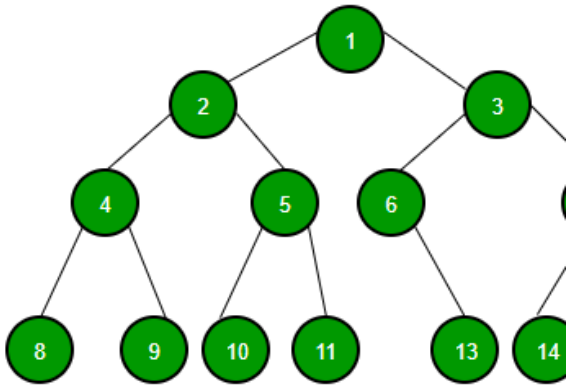


Figure 4. Generic Binary Tree

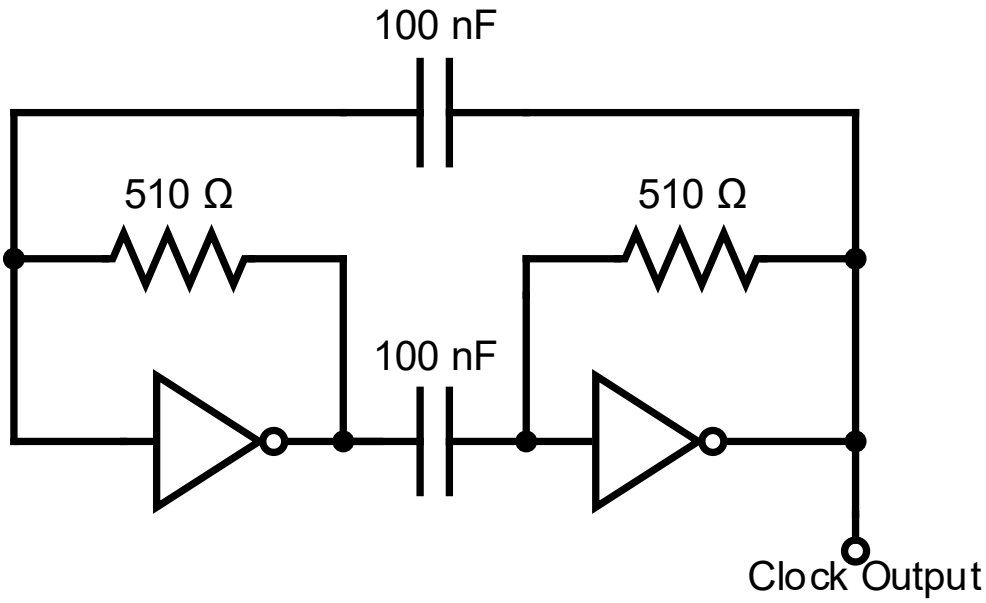
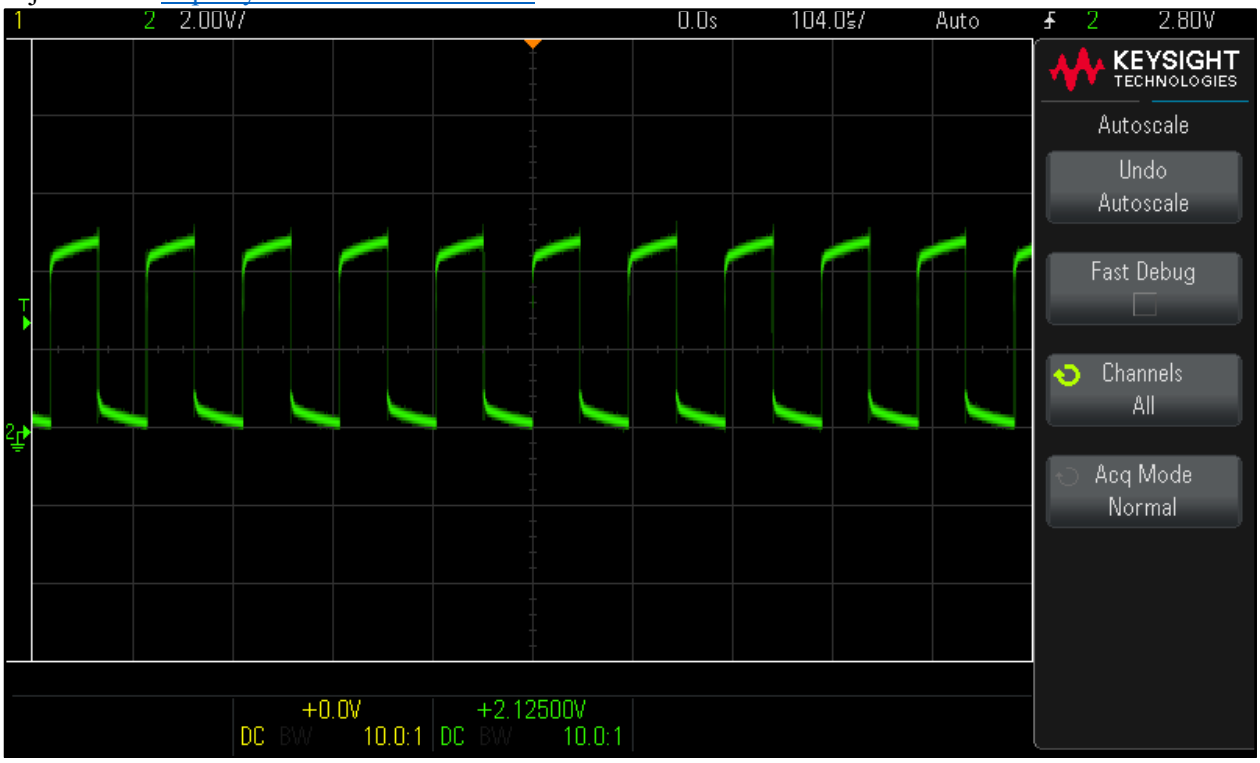


Figure 5. SAR ADC Clock Generator

The clock signal of the SAR ADC relies on an astable multivibrator built from NOT gates (see Figure 5). In this configuration, the resistors create a feedback loop between each NOT gate. As the capacitors constantly charge and discharge, a square wave is created. The clock speed of the SAR ADC is approximately 10 KHz, with a period of approximately 103 µs (see Media section).

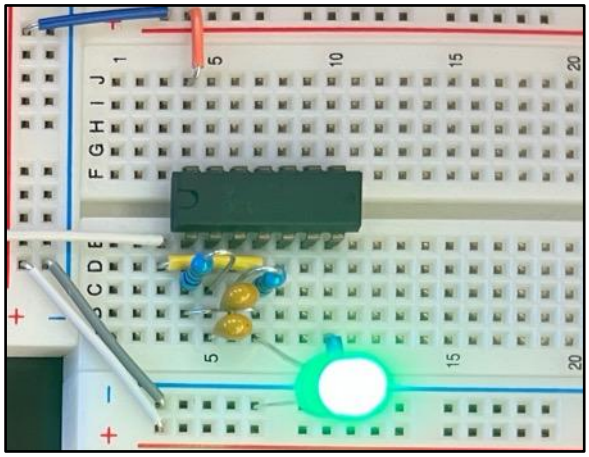
Media

Project video: https://youtu.be/dIuDrUOZVYI



Oscilliscope Reading of Clock Circuit (Period of ~103 µs)

DMM Confirmation of ~10 KHz



Clock Oscillator Circuit from 74HC04

Reflection

This is a very satisfying introduction to the SAR ADC. Conceptually, while I had a general idea of how successive approximation worked, I now have a solidified understanding. I think that it is not a difficult concept to grasp, in fact, I would argue that this is the simplest project we have had (conceptually) since grade 10. I will admit that building the circuit will likely prove to be slightly cumbersome and require much troubleshooting (good thing:) that I am looking forward to!

Project 3.5.2 R/2R Ladder DAC

Purpose

The purpose of the utilisation of an R/2R ladder DAC in the SAR ADC is to provide a variable reference voltage to compare the sample voltage to on each iteration of the successive approximation.

Reference

Project description: http://darcy.rsgc.on.ca/ACES/TEI4M/SARADC/index.html#DAC

Theory

An operational amplifier, or op amp, is an extremely versatile electronic component that is made from resistors, capacitors, and transistors (see Figure 1). On use of an op amp is a comparator, which is a device that is used to compare to voltage levels. In this configuration, when the voltage presented on the non-inverting input exceeds that presented on the inverting input, the output of the op amp is high. Therefore, when a known threshold is presented on the inverting input, the level of an unknown voltage relative to the threshold can be tested. This principle is extremely important for the operation of the SAR ADC. In the SAR ADC, an unknown voltage is successively approximated by changing the threshold present on the inverting input.

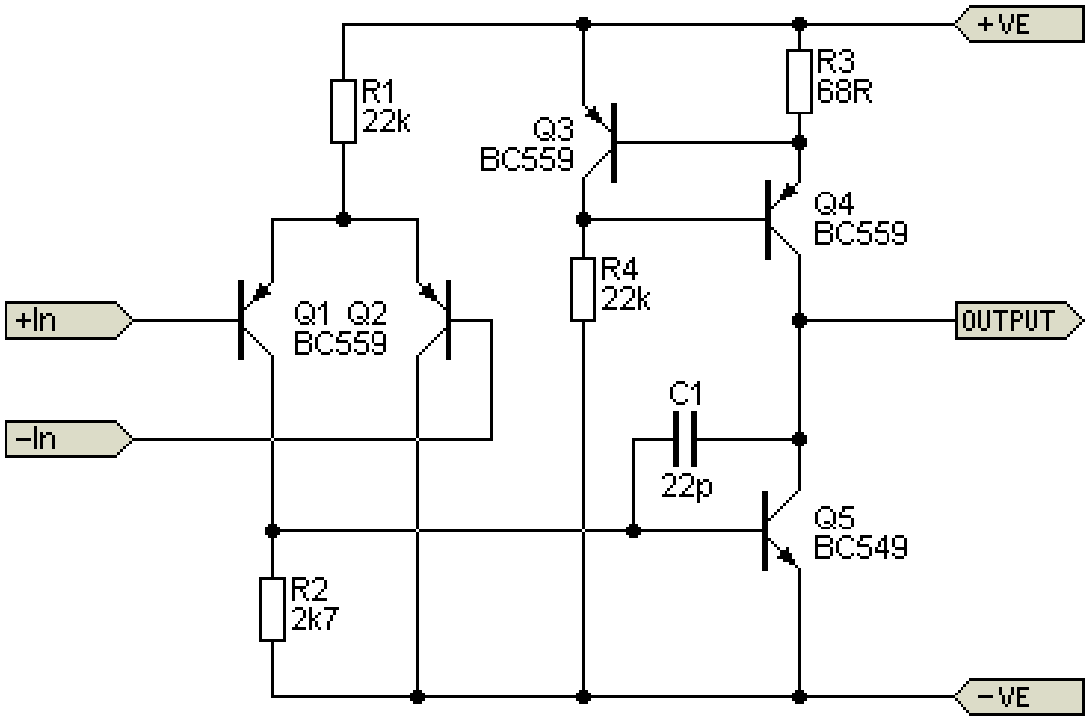


Figure 1. Operational Amplifier Internals

Procedure

The LF398 sample and hold IC utilized in the SAR ADC (see Project 3.5.1:Procedure) retains inputs that fluctuate between 0 and 5 V. To do this, it requires a negative and positive power supply of 9 V. To obtain this voltage range, two standard 9 V power supplies are utilized. In this configuration, the two power supplies are wired in series, and a virtual ground is established where the positive and negative terminals meet (see Figure 1). The positive terminal of the first power supply is 9 V higher that the virtual ground, so it is positive 9 V. The negative terminal of the second power supply is 9 V lower than the virtual ground, so it is negative 9 V.

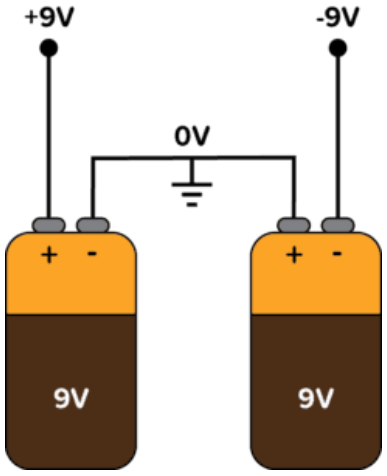


Figure 1. ± 9 V Power Supply

A supply range greater than 5 V is required because a buffer above and below the operating range is needed. This is because the LF398 is not a *rail-to-rail* component. A rail-to-rail component is a component in which the output can swing all the way to either the positive or negative (or ground) supply rail. The LF398 utilizes bipolar transistors, which consume some voltage. This means that without a dual supply of ±9 V, the held voltage would differ from the sampled voltage.

The sample and hold IC requires a storage capacitor to store the voltage that has been sampled for the ~1 ms that it takes for the voltage sample to be converted to a digital value. In the final stage of the SAR ADC, a 1000 pF capacitor is utilized; in this stage of the SAR ADC, however, a 100 nF capacitor is present. This is because a 100 nF capacitor holds the sampled voltage for longer, allowing the user to view the internal workings of the SAR ADC in real time. In the final version however, a quick charge and discharge is preferred since the sampling process completes in under 1 ms.

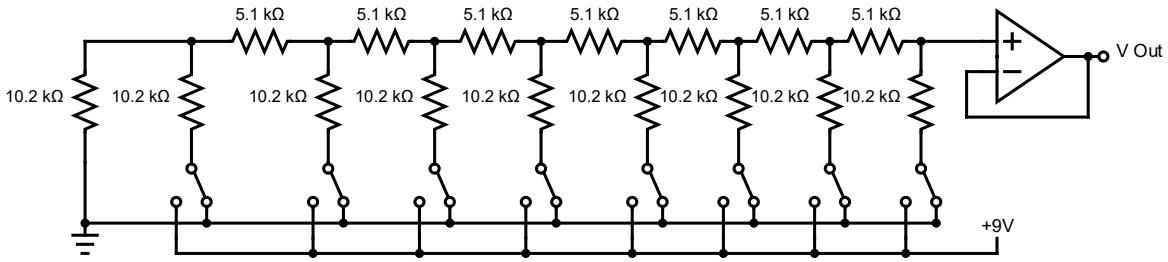


Figure 2. R/2R DAC with an Operational Amplifier-Driven Output

An R/2R DAC is a digital to analog converter that operates on the principle of voltage division (see Project 2.1 and Project 3.3) with binary-weighted inputs. Figure 2 depicts such a device, with. In this configuration, the MSB is represented by the switch on the right. This is because each bit to the left becomes progressively closer to the negative rail, bringing its overall contributed voltage closer to the negative rail. The resistor values are selected in a fashion that causes the inputs to be binary-weighted.

Figure 3 depicts the process that takes place in this configuration: the user dials the potentiometer to an unknown voltage, and stores it into the sample and hold IC by pressing the PBNO. Then, the user turns on the MSB. If the LED remains on, the user moves to the next switch, and repeats the process. If not, the user resets the MSB before continuing. This models the process that the eventual conversion process that will take place.

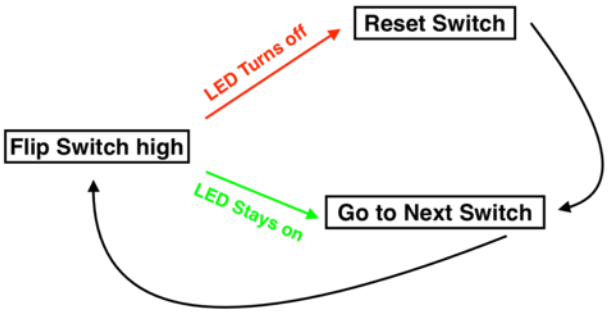


Figure 3. Manual SAR Process

Since the entire SAR ADC operates at 5 V (with the exception of the sample and hold IC and the comparator), a total of 3 voltage levels are required: 5 V, 9 V, and -9 V. Each 9 V rail is supplied by a discrete switching power supply.

The 5V rail, however, is supplied by an LM7805 voltage regulator (see Figure 4). In this configuration an LM7805 converts power from the 9 V rail to 5V by turning 4 V into heat.

This (inefficiently obtained) 5 V is then distributed to the parts that require it. The 9 V power supply can still be used in parallel for the comparator and the sample and hold IC.

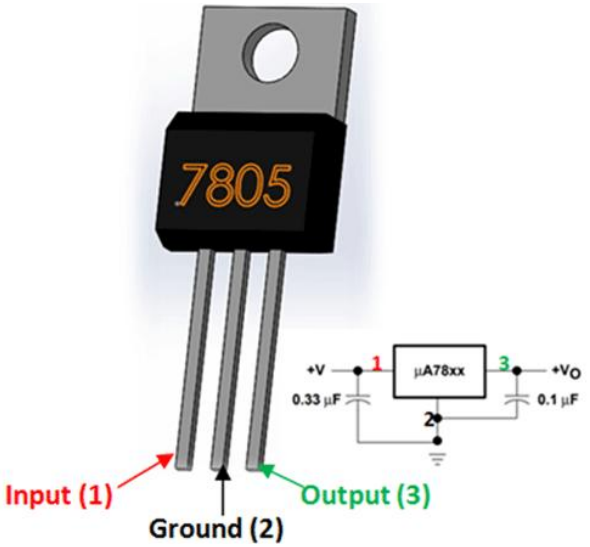


Figure 4. Manual SAR Process

This stage of the SAR ADC is asynchronous, meaning that it does not rely on a clock signal. The output of the R/2R DAC is fed into the sample and hold IC, who's output is used as the lower threshold of the comparator (see Theory section). This stage of the SAR ADC allows the user to manually sample and convert a voltage using a DIP switch bank, a potentiometer, a PBNO, and an LED.

The LM393 has an *open-collector* output. An open-collector means that the output can only sink current, and is unable to source current. To interface with digital logic circuits, the output is pulled high (see Figure 5).

There are several advantages of an open-collector system: firstly, it can interface with different voltage levels. For example, the comparator can operate at 5 V but pull up the output to easily operate at 3.3 V. Additionally, multiple open-collector outputs can easily be wired together to function with AND gate logic.

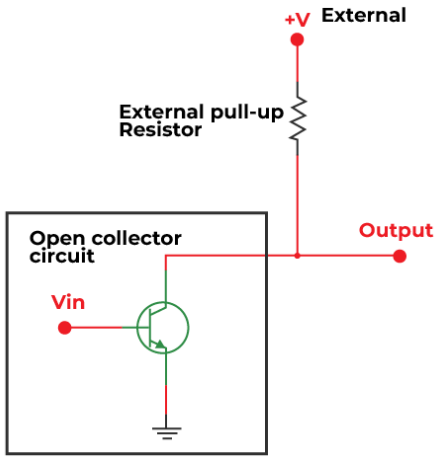
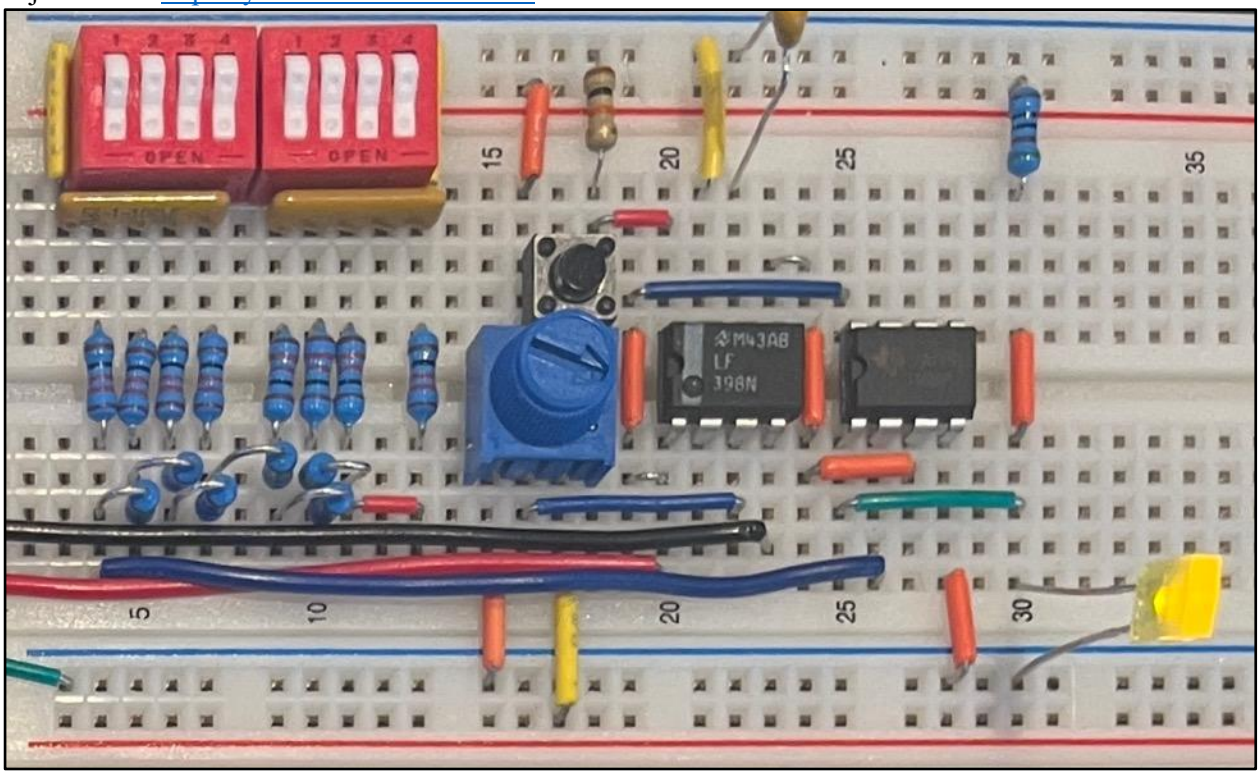


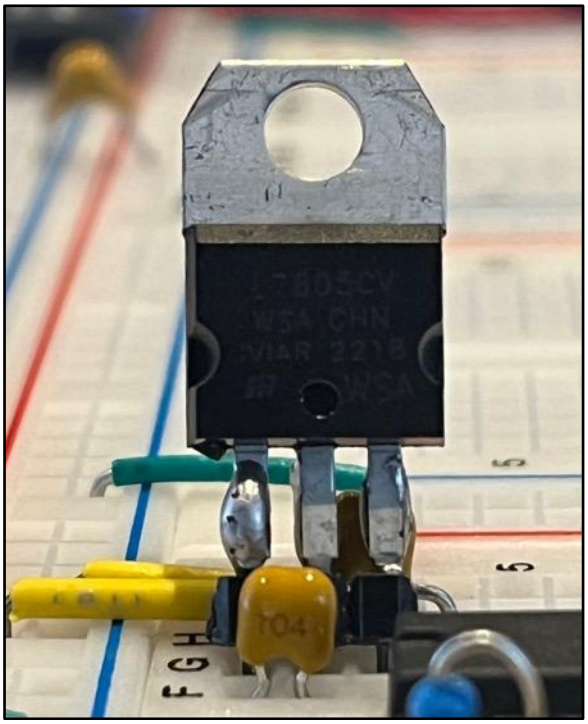
Figure 5. Open-Collector Output Configuration

Media

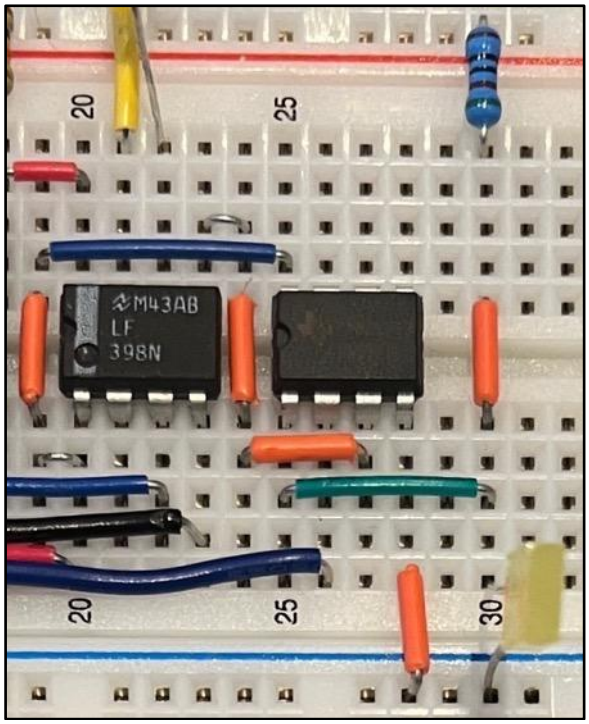
Project video: https://youtu.be/KrAzBvUE23o



SAR ADC R/2R DAC, Sample and Hold IC, and Comparator



LM7805 Voltage Regulator and Decouling Caps



Comparator and Sample and Hold IC Close-up

R/2R Ladder Closeup

Reflection

This has been an okay stage of the project. I feel that this stage could have been combined with the overview and clock, as I wrote about much of the R/2R ladder and comparator stage within the overview. Other than that, I am pretty satisfied with the way that the analog input works. I am able to replace the SAR logic with my own brain by flipping the switches which is excellent for understanding how it works. I also was introduced to negative voltage in this project. Even though we were supposed to include negative voltage in the first R/2R project, I never did, but now I understand it.

I also am really appreciating the importance and versatility of the R/2R ladder circuit. In grade 11, I thought it was kind of cool, but not necessarily a practical or useful circuit. At this point, I have used it in two additional projects: my ISP (which relied on it very heavily) and the SAR ADC (which also relies on it quite heavily). Overall, my prototype seems to work quite well thus far, and I'm excited to finish the project!

Project 3.5.3 SAR ADC Completed

Purpose

The purpose of the completed SAR ADC is to convert analog voltages to a discrete digital value between 0 and 127. The current voltage being read can also be accessed over an I2C bus; this ADC is capable of acting as an I2C peripheral in both binary and voltage-reporting mode.

Reference

Project description: http://darcy.rsgc.on.ca/ACES/TEI4M/SARADC/index.html#Completed

Build credit: https://hackaday.io/project/181826-homemade-successive-approximation-register-adc

Original author: Mitsuru Yamada

Theory

One of the main measures of an ADC's processing abilities is its sample rate. The sample rate is the maximum frequency at which the ADC can take samples of its input. The SAR ADC has a clock frequency of 10 KHz, and each sample takes 8 clock cycles, the sample rate is given by:

$$f = \frac{10\ \text{KHz}}{8\ \text{Samples}} = 1.25\ \text{kS/s}$$

This means that in 1 second, the SAR ADC takes approximately 1250 samples of the voltage presented on its analog input. This sample rate can easily be increased by increasing the clock frequency. While changing the bitness (see Project 3.2.5) of an ADC requires major changes in the wiring and components of the circuit, the clock frequency requires little change, as long as the components can handle a higher clock frequency.

Figure 1 depicts the difference between resultant wave functions with progressively higher sample rates. Increasing the sample rate even with a low bit-depth ADC can allow its output to much more closely resemble the original waveform (see Figure 1).

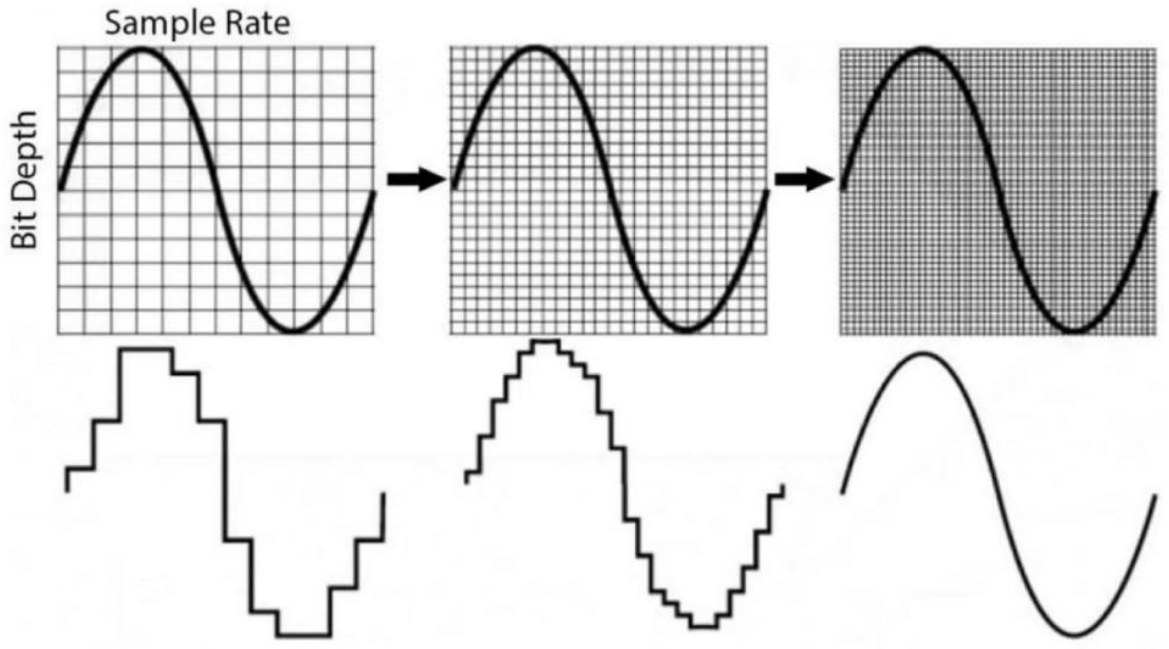


Figure 1. Sample Rate Increase vs Bitness Increase

Procedure

The sample rate of 1.25 KHz is relatively fast, although it is not nearly fast enough for many audio-based applications which would require a minimum frequency of ~16 KHz. To calculate the maximum theoretical clock frequency that the SAR ADC could support while maintaining proper operation, the frequency-limiting IC must be found. The logic gates and flip-flops have propagation delays in the range of ~10ns, whereas the sample-and-hold IC has a propagation delay of ~8 µS with a 1 nF capacitor (see Figure 1). Assuming the SAR takes around 2 µS to compete the successive approximation, 1 sample would take a total of 10 µS. This translates to a theoretical sample rate of 1 MHz, or a clock frequency of 8 MHz.

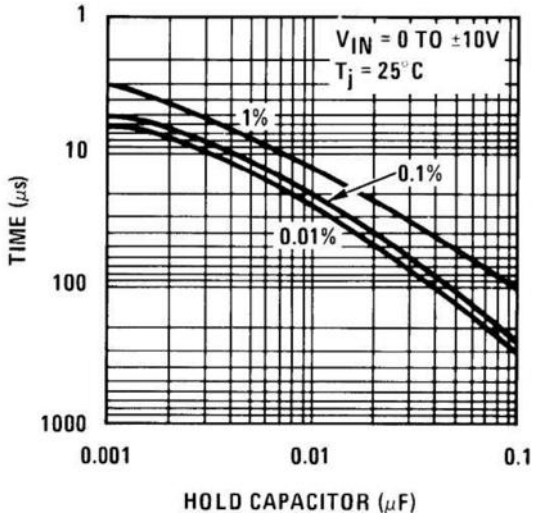


Figure 1. LF398 Acquisition Time

The SAR ADC supports two modes: manual, in which the output is latched and only updated when the sample button it pressed, and automatic mode, in which the output constantly updates to match the input voltage, at a sample rate of 1.25 KHz.

In this stage of the SAR ADC, the manual input to the R/2R Ladder is replaced with AND logic (see Figure 2). In this configuration, 1 input of the AND gate is connected to a D flip-flop, while the other is connected to a bank of cascaded D flip-flops. These flip-flops (not depicted in Figure 2) are cascaded from MSB to LSB, with the MSB connected directly to power. This means that on the first clock cycle, only the MSB has the chance for its output to be high. Then, on each successive clock cycle, the next bit's cascaded D flip-flop goes high, allowing it to update bit-by-bit.

The data input of the D flip-flops depicted in Figure 2 is connected to the comparator, and the clock is connected to the next bit's cascaded D flip-flops, which causes it to latch to the comparator's output. This is how the SAR ADC sets each bit to the correct value; when the bit's threshold is exceeded, the comparator is high, and latched to the appropriate flip-flop. This configuration automatically updates the R/2R ladder for the next iteration.

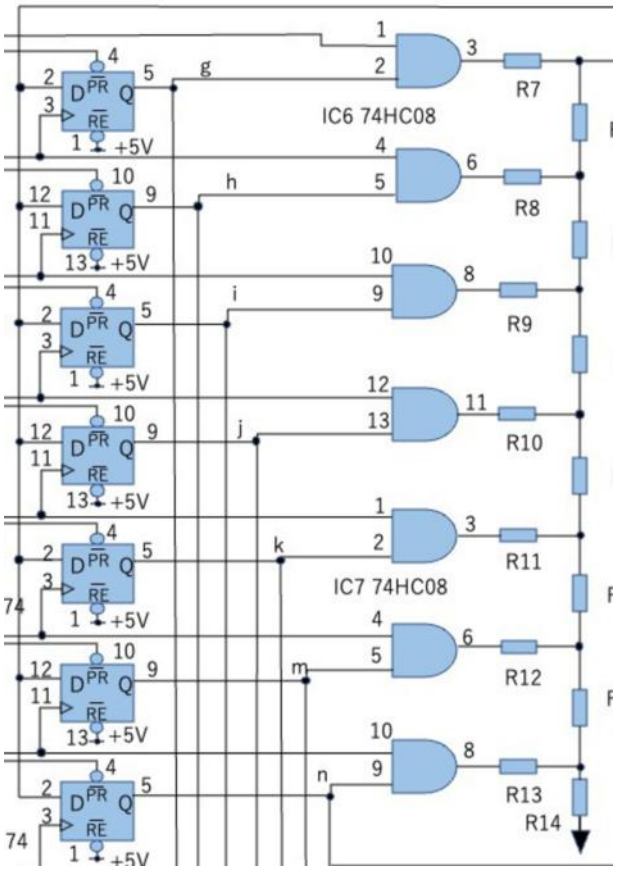


Figure 2. R/2R Ladder Inputs

This version of the SAR ADC is I2C compatible, meaning the sampled value can be accessed over an I2C bus. This is a direct result of the addition of an Arduino Nano in I2C slave mode, creating a smart I2C network (see Figure 3). In this configuration, the Nano has the outputs of the SAR ADC connected to digital pins 0-6, which is most of PORTD. Then, when the I2C master requests the data, it downloads the value in PIND, reading the entire SAR ADC output. The data can be requested in 3 different ways: 7-bit binary mode, 8-bit binary mode, or floating-point voltage mode.

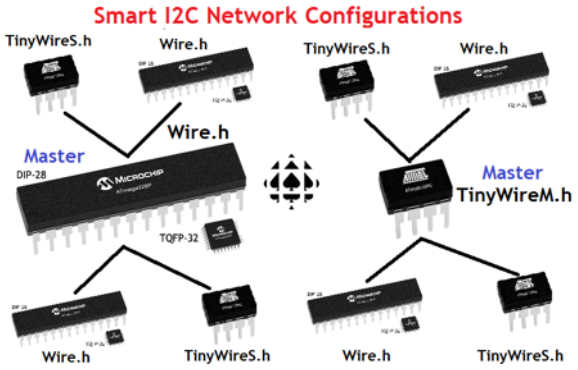


Figure 3. Smart I2C Network

7-bit binary mode sends the raw data, because the value read from PIND is an 8-bit value, but the MSB is not utilized, making it a 7-bit value. In 8-bit binary mode, the slave automatically bit shifts PIND left 1 place before sending it. This is because 8-bits is a standard number of bits, while 7 is not. By automatically converting the value before sending it, the master can use the raw received data, saving it clock cycles. When floating-point voltage mode is active, the slave automatically converts the binary value to a voltage value between 0 and 5 V with the following equation, where S is the 7-bit sample:

$$V = \frac{S}{127} \times 5\ \text{V}$$

I2C is an 8-bit protocol, while a float is 32-bit datatype with a distinct structure (see Figure 4). A float is interpreted in a similar fashion to scientific notation. The MSB is the sign, with a 0 corresponding to a positive value, and a 1 corresponding to a negative. The next 8 bits are the exponent, which is how many times the mantissa must be shifted left or right. Finally, the mantissa is the actual value, with 1 leading bit.

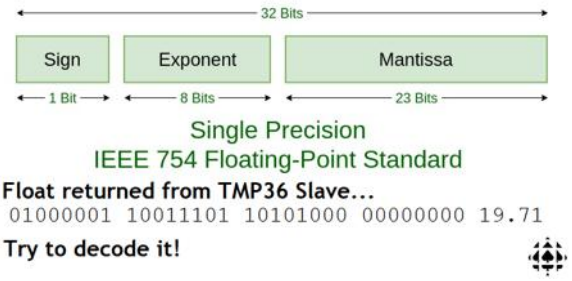


Figure 4. Single-Precision Float Structure

In order to send this value of I2C, a *union* is used. A union is a C programming structure in which 2 variables occupy the same address in memory. In this configuration, a 4-byte chunk of RAM can be written to as an array of 8-bit values, but then read from (and interpreted as) a floating-point decimal.

The slave utilizes a union to convert the float to a 4-location array of 8-bit values. It then sends these bytes over I2C, and the master has a union to reassemble this array into the floating-point number (see Figure 5). The master then prints the voltage over the serial monitor, allowing the value read to be graphed as a function of time.

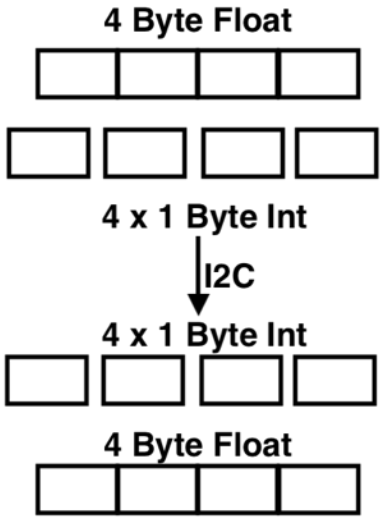


Figure 5. Float I2C Send Structure

To request multiple data points, a command code system is implemented. In this configuration, the master sends a code corresponding to a data point, which is placed into a buffer. Then, the master sends a request. Within the request routine, a subroutine is included for each point (see Figure 6).

Figure 6. Code Data Correspondences	
Code	**Mode**
0	7-Bit
1	8-Bit
2	Floating-Point Voltage

While accessing this SAR ADC over I2C does not have any major benefits over using the built in ADCs through `analogRead()`, it is simply a proof of concept. A higher frequency, higher bit-depth SAR ADC can be built in a similar fashion. In this configuration, the output could be 10 bits or more. Utilizing the main MCU to read from this SAR ADC is not practical; it would consume too many I/O pins. With an embedded I2C controller (an Arduino Nano, in this case), only two pins are consumed, and some data processing can be automatically done (see Figure 7).

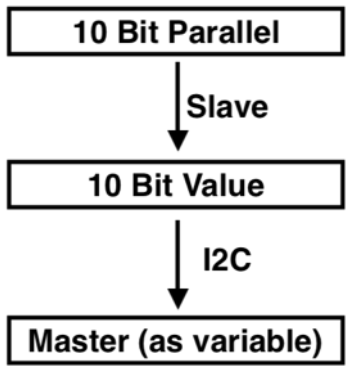


Figure 7. Float I2C Send Structure

It also allows for general expansion of the master. If extra EEPROM is needed, the master can offload the data to the slave's built in EEPROM. If extra (slower speed) RAM is needed, the master can store it in the slave's extra RAM.

Additionally, multiple ADC pins can be read by a singular I2C controller. In this configuration, the input of the SAR ADC is connected to an analog multiplexer, with several input pins, and a select line is controlled by the slave (see Figure 8).

The slave then expects a byte to be sent which corresponds to the ADC number (equivalent to A0-A5), and appropriately configures the multiplexer select line, reading from the appropriate input pin.

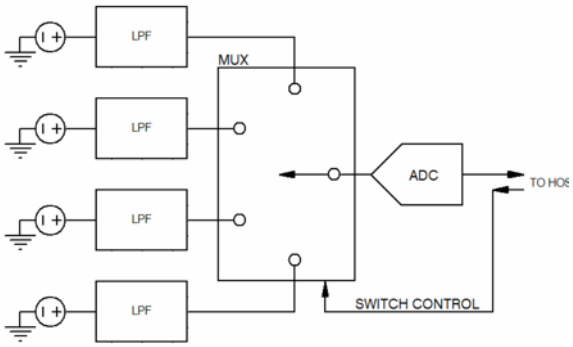


Figure 8. Multiplexed ADC

Code

Master

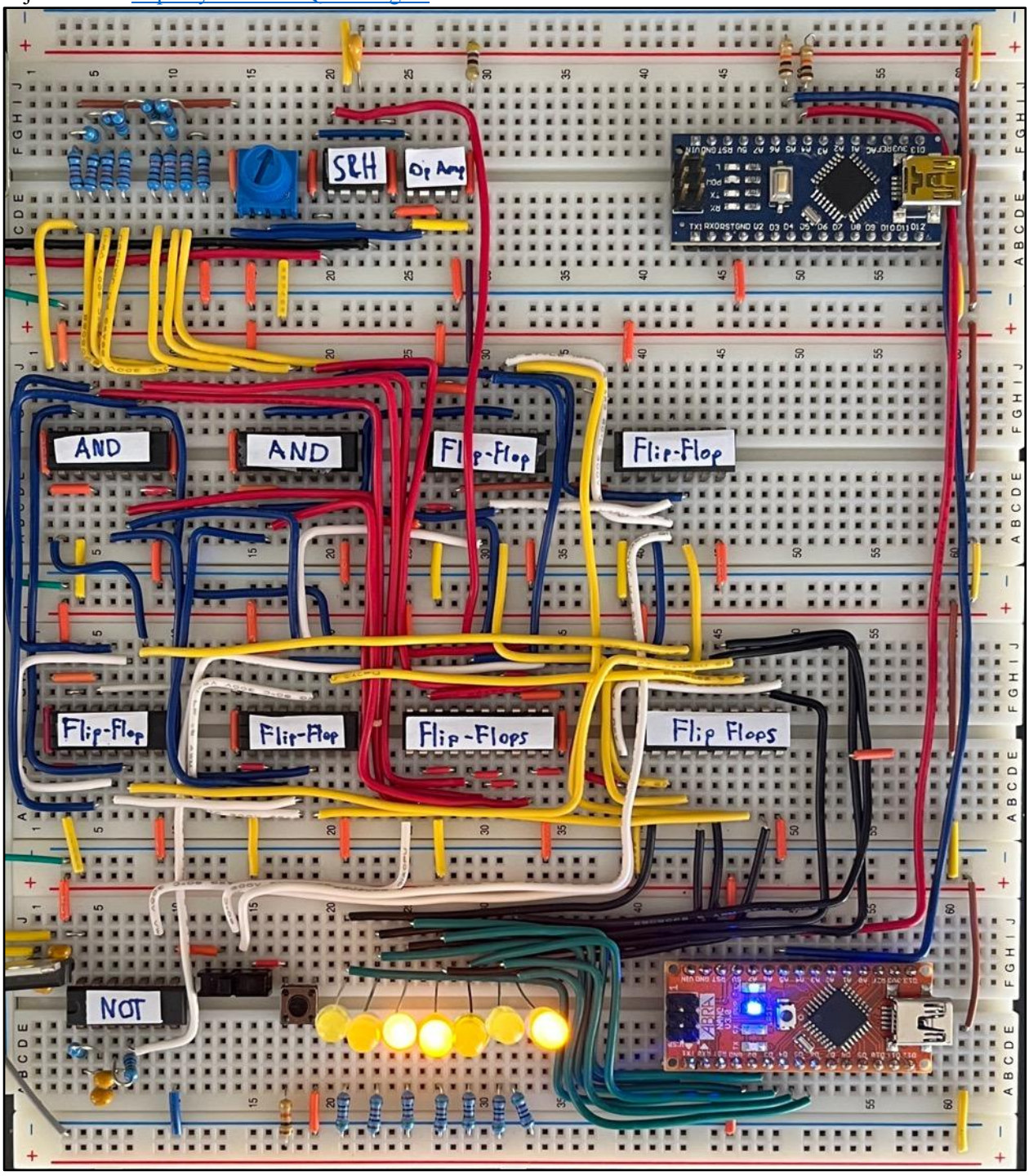
```
// PROJECT   : SAR Master
// AUTHOR    : R. Jamal
// PURPOSE   : To plot the I2C-compatible SAR ADC readings
#include <Wire.h>             //Include Wire for I2C
#define ADDR 0x20             //Slave I2C address
void setup() {
  Wire.begin();                 //Initialize I2C bus
  Serial.begin(9600);
}
void loop() {
  Wire.beginTransmission(ADDR); //Begin transmission to slave
  Wire.write(2);                //Request voltage on request
  Wire.endTransmission();       //End transmission
  Wire.requestFrom(ADDR, 4);    //Request 4 bytes
  while (!Wire.available());    //Wait for data to become available
  union b2f {                   //Union to receive float
    uint8_t b[4];               //8-bit wide array of 4
    float f;                    //Float componenent
  } data;                       //Name of union
  data.b[3] = Wire.read();      //Reads first byte
  data.b[2] = Wire.read();      //Reads second byte
  data.b[1] = Wire.read();      //Reads third byte
  data.b[0] = Wire.read();      //Reads fourth byte
  Serial.println(data.f);       //Print the received voltage
  delay(10);                    //Wait 10 ms per reading
}
```

Slave

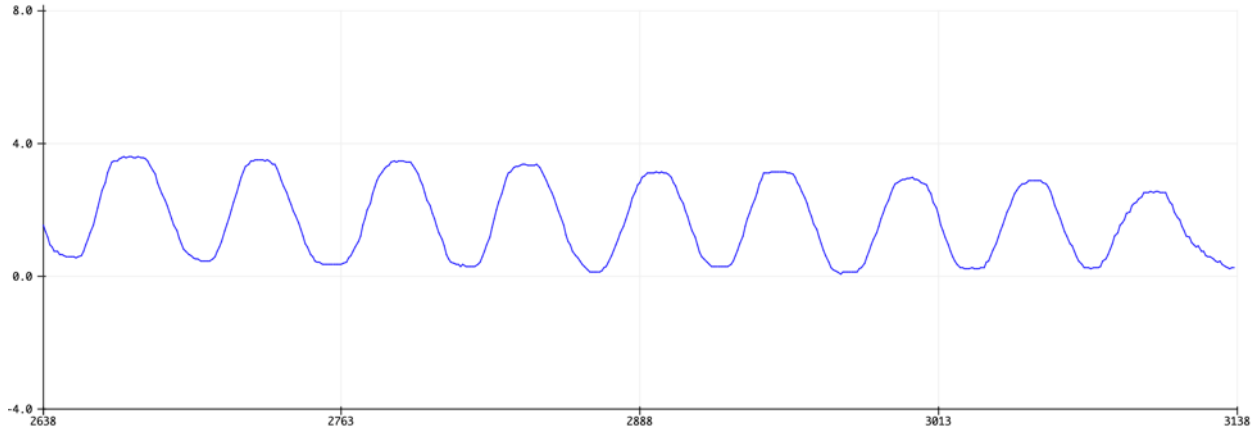
```
#include <Wire.h>                       //Include Wire for I2C
#define ADDR 0x20                       //Slave I2C address
uint8_t code;                           //Buffer for datatypes
/*     Codes:
   0 - Binary 8-bit
   1 - Binary 7-bit
   2 - Full voltage
*/
void setup() {
  Wire.begin(ADDR);                     //Initialize I2C
  Wire.onReceive(ISR_receive);          //Define receive handler
  Wire.onRequest(ISR_request);          //Define request handler
}
void loop() {}                          //Nothing to do
void ISR_receive(uint8_t bytes) {       //Receive handler
  code = Wire.read();                   //Receive datatype definition
}
void ISR_request() {                    //Request handler
  if (!code) Wire.write(PIND << 1);     //Report 8-bit value
  else if (code == 1) Wire.write(PIND); //Report 7-bit value
  else {
    union b2f {                         //Union to send float
      uint8_t b[4];                     //8-bit wide array of 4
      float f;                          //Float componenent
    } data;                             //Name of union
    data.f = (float(PIND) / 127) * 5;   //Calculate voltage
    Wire.write(data.b[3]);              //Sends first byte
    Wire.write(data.b[2]);              //Sends second byte
    Wire.write(data.b[1]);              //Sends third byte
    Wire.write(data.b[0]);              //Sends fourth byte
  }
}
```

Media

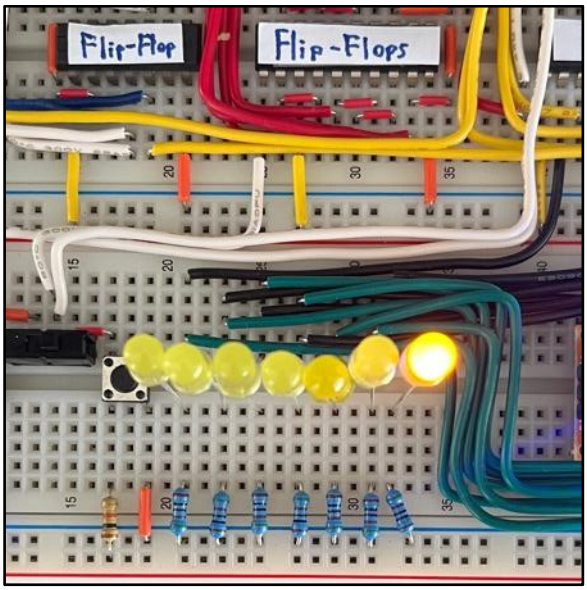
Project video: https://youtu.be/NQ83lcEegKc



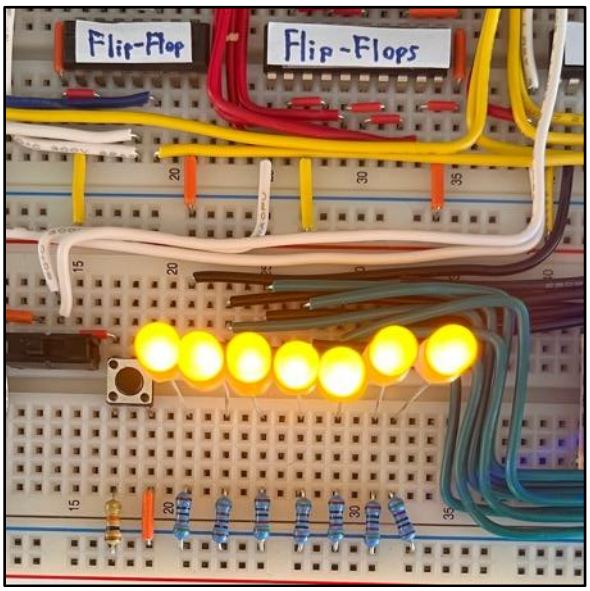
SAR ADC Entire Build



Serial Plotter when Potentiometer is Spun Back and Forth in Floating-Point Voltage Mode



SAR ADC Output at ~2.5 V



SAR ADC Output at ~5 V

Reflection

This has actually been quite a good project. At the beginning, I must say that I was somewhat skeptical, and not sure whether or not it would be a project that I thoroughly enjoyed. Like CHUMP, it was my addition to the project that is what really sparked my interest. When I integrated the project with a microcontroller, I began to see how cool it really was. I was mostly just surprised to see how accurate it was. When I connected my DMM to the input, and put the slave in floating point voltage mode, they showed the exact same value which was pretty cool. I also was really surprised to have found a use for unions so quickly. When we learned about them in class, I did not think I would have the chance to use them so quickly, but my project certainly would not have been possible without their use.