This next video will explain how a typical ADC works. Again the goal of an analog to digital converter is to take an analog signal (0 to 3.3 V on our microcontroller) and convert it to a digital number (0 to 4095 on our microcontroller).

VIDEO 14.2. SUCCESSIVE APPROXIMATION

C14 Video 2 Successive Approximation YouTube

4:45 / 4:45 1.0x

DR. RAMESH YERRABALLI: So in this video, we

will show you the essential process by which the analog to digital conversion

occurs.

This is called successive approximation.

DR. JONATHAN VALVANO: There are a couple of components.

The unknown input exists here that we're trying to sample,

and will eventually show up as a digital output.

We're going to demonstrate how this works by playing a game.

And we need one person to be mean and one person to be smart.

Professor Yerraballi, which one do you want to be,

the mean one or the smart one?

DR. RAMESH YERRABALLI: If you ask like that, I want to be the smart one.

DR. JONATHAN VALVANO: OK.

So I'm the mean person.

That's me right here.

And what has just happened is I have guessed a number, an 8-bit number.

And this 8-bit number can exist from zero to 255,

but I'm not going to tell you what it is.

You're going to have to guess.

DR. RAMESH YERRABALLI: But you're going to give me some hints.

DR. JONATHAN VALVANO: I am.

So Professor Yerraballi is going to make the guess right here.

And then I'm going to tell him whether his answer is high or low.

DR. RAMESH YERRABALLI: OK.

I'm going to guess 128.

DR. JONATHAN VALVANO: OK.

Good guess.

Professor Yerraballi guesses 128.

My answer is lower than that.

DR. RAMESH YERRABALLI: So it's lower than that.

So I'm going to choose 64.

DR. JONATHAN VALVANO: Ah, good guess.

64.

But not right.

My answer is lower than 64.

DR. RAMESH YERRABALLI: OK.

Even lower.

So I'm going to guess 32.

DR. JONATHAN VALVANO: Ah, 32.

Well, now you've stumped me.

My guess is actually higher, or it might be 32.

DR. RAMESH YERRABALLI: OK.

So I will guess between 64 and 32.

I'll guess 48.

DR. JONATHAN VALVANO: Hm.

48.

My guess is lower than 48.

DR. RAMESH YERRABALLI: OK.

So 32 and 48.

I guess 40.

DR. JONATHAN VALVANO: Ah.

You're getting closer.

My guess is higher than 40, or it might be 40 itself.

DR. RAMESH YERRABALLI: OK.

So how about 44?

DR. JONATHAN VALVANO: Ah.

Closer still.

My guess is lower than 44.

DR. RAMESH YERRABALLI: Go on.

Give me your answer.

I'm going to go for 42.

DR. JONATHAN VALVANO: Hm.

42.

My answer is higher than 42, or it might be 42 itself.

DR. RAMESH YERRABALLI: OK.

How about 43?

DR. JONATHAN VALVANO: My answer is lower than 43.

Help

DR. RAMESH YERRABALLI: Ah.

You got to be kidding me.

It is 42.

DR. JONATHAN VALVANO: It is!

Good guess.

My answer was 42.

And that's what will come out over here.

Do you want to see a cool trick?

DR. RAMESH YERRABALLI: Yeah, show me.

DR. JONATHAN VALVANO: All right.

What if every time you guessed, I gave a binary value?

What if I said low meant 0 and high meant 1?

So you guessed low, low, high, low, high, low, high, low, in that order.

DR. RAMESH YERRABALLI: Ah.

So what you've done is in each guess you're

guessing one bit of the binary representation of the number.

DR. JONATHAN VALVANO: Absolutely.

DR. RAMESH YERRABALLI: Starting with the most significant bit

and winding down to the least significant.

DR. JONATHAN VALVANO: Yeah.

So if you look at this first guess, we can

see something very significant happening.

By guessing 128, I was forced to tell Professor Yerraballi what

bit seven was, and I did that by saying it was low.

Then, knowing bit seven, he then guessed bit six.

And one by one, he guessed each bit in order from bit seven all the way down

to bit zero.

DR. RAMESH YERRABALLI: That's a cool trick.

DR. JONATHAN VALVANO: It's not really a trick.

It's actually how your A to D converter in your microcontroller works.

It's called successive approximation because each guess reveals one bit.

Help

The most pervasive method for ADC conversion is the **successive approximation** technique, as illustrated in Figure 14.2. A 12-bit successive approximation ADC is clocked 12 times. At each clock another bit is determined, starting with the most significant bit. For each clock, the successive approximation hardware issues a new "guess" on V_{dac} by setting the bit under test to a "1". If V_{dac} is now higher than the unknown input, V_{in}, then the bit under test is cleared. If V_{dac} is less than V_{in}, then the bit under test is remains 1. In this description, *bit* is an unsigned integer that specifies the bit under test. For a 12-bit ADC, *bit* goes 2048, 1024, 512, 256,...,1. D_{out} is the ADC digital output, and Z is the binary input that is true if V_{dac} is greater than V_{in}.

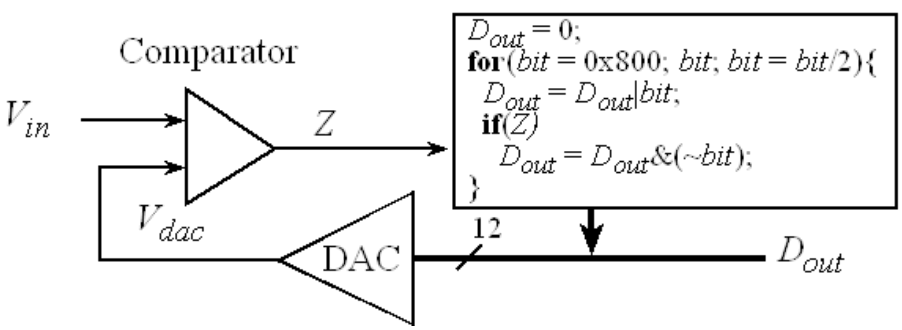

Figure 14.2. A 12-bit successive approximation ADC.

Observation: The speed of a successive approximation ADC relates linearly with its precision in bits.

Normally we don't specify accuracy for just the ADC, but rather we give the accuracy of the entire system (including transducer, analog circuit, ADC and software). An ADC is **monotonic** if it has no missing codes as the analog input slowly rises. This means if the analog signal is a slowly rising voltage, then the digital output will hit all values one at a time, always going up, never going down. The **figure of merit** of an ADC involves three factors: precision (number of bits), speed (how fast can we sample), and power (how much energy does it take to operate). How fast we can sample involves both the ADC conversion time (how long it takes to convert), and the bandwidth (what frequency components can be recognized by the ADC). The ADC cost is a function of the number and quality of internal components.

Two 12-bit ADCs are built into the TM4C123/LM4F120 microcontroller. You will use ADC0 to collect data and we will use ADC1 and the PD3 pin to implement a voltmeter and oscilloscope.