In the next video we will show the software to initialize and sample the ADC.

CAPTURING A SAMPLE

C14 Video 4 Software on the TM4C to sample one channel YouTube

Help

4:54 / 4:54 1.0x

DR. RAMESH YERRABALLI: So we looked at the two steps.

One is the initialization step, which is what this routine is doing.

The initialization ritual, which we do one, which is a 13 step sequence.

You can look at all the details here.

We talked about it.

This part here is simply making the GPI pin be an alternate pin and an input.

So this is saying input.

We did the clock.

We did the alternate bit.

And then we did select, and we enabled (Disable digital) it.

Now, the second part is we turn on the clock for the ADC.

We wait for a little bit too for the clock to get going.

We set the sampling rate.

We set the priority of the sequencer.

We disable it and enable it between these two steps.

And while we are configuring it, the steps involved

are, what is the trigger we're using?

We're using software as a trigger.

We are saying, I'm using channel PE2, which is analog in one(AIN1).

And we made sure that IE0 is set to one.

DR. JONATHAN VALVANO: I have a question, Professor Yerraballi.

Why did you disable the digital input?

DR. RAMESH YERRABALLI: Oh, that is correct.

In fact, what we did for the D-enable is we turned it off.

DR. JONATHAN VALVANO: Yeah, but why?

DR. RAMESH YERRABALLI: That is because it's not digital.

It's analog.

DR. JONATHAN VALVANO: Ah, yes.

You're right.

Help

DR. RAMESH YERRABALLI: All right.

So let's take a look at our second routine, which is our routine

where we're actually going to be read the data.

So this is a data read routine.

This is when there is a sample, and you're ready to read it.

And this routine has been called.

This routine's responsibility is to tell the ADC module

that you are ready to read it, which is our first step.

OK.

Let's take a look at the read routine.

We've already seen it, so I'm going to summarize it.

We start.

And once we get the sampling going, we'll

keep checking to see if it is done.

So you're we're going to check the status bit.

And if the status bit says that it's Busy,

then we keep going back and keep checking it.

And eventually, it's going to say it's done.

And once it's done, we're going to read the data.

And we're going to clear the bit.

So that's exactly what you see here.

This is a start.

We have a loop here, which repeatedly checks the status.

Then we come out of this.

We read the data, which is right here in the FIFO3.

And the last step is we cleared the bit, which is right here.

And we return the results.

So we return.

DR. JONATHAN VALVANO: All right.

Shall we see if it works?

DR. RAMESH YERRABALLI: Let's do it.

DR. JONATHAN VALVANO: All right.

So we have a analog signal connected up to PE2.

So to test it, we ill use this main program.

And we will ask the ADC converter to capture the input

and store it into this variable.

All right.

Let's go.

Build.

Download.

Debug.

Let's look at the watch window.

And in this watch window, we have the variable

that we're going to set every time we sample.

So let's hit the Go button.

In this watch window, we can see the results of the ADC converter.

That's a 12-bit number.

So Professor Yerraballi, make it go smaller.

DR. RAMESH YERRABALLI: So I'm sliding the slide-pot.

And it's moving it to the right.

And I moved it as right as I can.

DR. JONATHAN VALVANO: Oh, that's a small number.

DR. RAMESH YERRABALLI: And it's a small

Help

number.

DR. JONATHAN VALVANO: OK.

The ADC converter goes from zero to,
what number does it go to?

Let's see.

Ooh, bigger.

DR. RAMESH YERRABALLI: Bigger.

DR. JONATHAN VALVANO: Bigger.

DR. RAMESH YERRABALLI: I'm moving it.

DR. JONATHAN VALVANO: Come on,
faster.

No, not so fast.

Slow it down.

There we go.

And the largest number is 4,095.

There we are, 12 bit converter.

Now, you try it.

Help

Program 14.2 gives a function that performs an ADC conversion. There are four steps required to perform a software-start conversion. The range is 0 to 3.3V. If the analog input is 0, the digital output will be 0, and if the analog input is 3.3V, the digital output will be 4095.

Digital Sample = (Analog Input (volts) • 4095) / 3.3V(volts)

Step 1. The ADC is started using the software trigger. The channel to sample was specified earlier in the initialization.
Step 2. The function waits for the ADC to complete by polling the RIS register bit 3.
Step 3. The 12-bit digital sample is read out of sequencer 3.
Step 4. The RIS bit is cleared by writing to the ISC register.

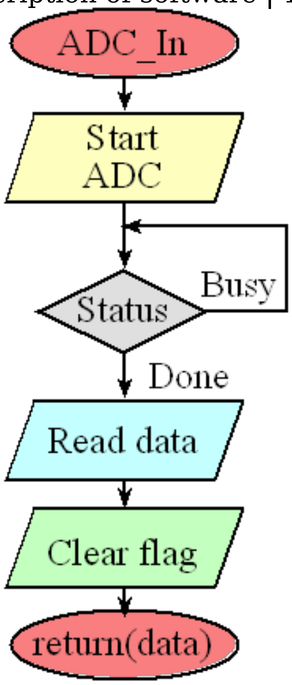

Figure 14.3. The four steps of analog to digital conversion: 1) initiate conversion, 2) wait for the ADC to finish, 3) read the digital result, and 4) clear the completion flag.

Help

```
//------------ADC_InSeq3------------
// Busy-wait analog to digital conversion
// Input: none
// Output: 12-bit result of ADC conversion
unsigned long ADC0_InSeq3(void){  unsigned long result;
  ADC0_PSSI_R = 0x0008;            // 1) initiate SS3
  while((ADC0_RIS_R&0x08)==0){};   // 2) wait for conversion done
  result = ADC0_SSFIFO3_R&0xFFF;   // 3) read result
  ADC0_ISC_R = 0x0008;             // 4) acknowledge completion
  return result;
}
```

Program 14.2. ADC sampling using software start and busy-wait (C14_ADCSWTrigger).

There is software in the book Embedded Systems: Real-Time Interfacing to ARM® Cortex™-M Microcontrollers showing you how to configure the ADC to sample a single channel at a periodic rate using a timer trigger. The most accurate sampling method is timer-triggered sampling (**EM3**=0x5). An example of timer-triggered ADC sampling can be found at the software download site for the book, ADCT0ATrigger_4F120.zip (http://users.ece.utexas.edu/~valvano/arm/ADCT0ATrigger_4F120.zip).

Unfortunately, the TExaS simulator does not simulate this important sampling mode, so you will not be able to use it for Lab 14. For Lab 14 you must use software triggered ADC sampling with sequencer 3.

CHECKPOINT 14.1

If the input voltage is 1.5V, what value will the TM4C 12-bit ADC return?

Hide Answer

Approximating the 12-bit ADC is linear, either Dout = 4096*Vin/3.3 or 4095*Vin/3.3 = 1862.

CHECKPOINT 14.2

If the input voltage is 0.5V, what value will the TM4C 12-bit ADC return?

Hide Answer

Approximating the 12-bit ADC is linear, either Dout = 4096*Vin/3.3 or 4095*Vin/3.3 = 621.