1. PROF.
2. RAMESH YERRABALLI: So in the last module,
3. we saw how stimuli in the external world can be captured by our computer
4. as long as it's rendered as a voltage.
5. So in this module, we'll look at how your micro-controller does this job.
6. That is, we'll look at the process of setting it up,
7. and then we'll look at the process of capturing the samples.
8. So on our micro-controller, which is our launchpad,
9. we have several of the GPIO pins that play dual roles.
10. So what we mean by the dual role is we'll choose their alternate function.
11. So there are 12 pins which are called 12 ADC pins which
12. are capable of receiving analog inputs, which are as rendered voltages,
13. and then converting them to 12-bit numbers using the ADC module on board.
14. So among these 12 ADC pins, two pins are the ones we
15. going to focus in this course.
16. One is PE2.
17. And the other is PD3.
18. This is referred to as Analog in 1.
19. And this is referred to as the Analog in 4.
20. By the way, we've already been using PD3 all along.
21. This is that analog pin that is used to measure voltages.
22. This was our way of measuring voltage on our circuit all along.
23. But AIN1, which is what we're going to use today,
24. is going to be used to measure voltages from an input device.
25. So let's get started on looking at all the device
26. registers that we have to manipulate to make ADC work.
27. So like any device that we've looked at, in order to manipulate the device,
28. we have to work with an initialization sequence.
29. The initialization sequence is done once,
30. or what we call as a set-up sequence.
31. And the setup sequence for ADC involves 13 steps.
32. I'm going to summarize the first few steps.
33. The first one through five steps are things that you already
34. area familiar with, which are make sure that the GPIO
35. pin-- here a GPIO pin for port E-- pin 2-- is properly configured.
36. So which means we turn on the clock.
37. We make it an input.
38. So we manipulated the direction register.
39. We enable its alternate function, which is the most important thing, which
40. is different from what we were doing so far.
41. So we enable its alternate function.
42. And then we (disable) it.
43. DEN, we (disable digital) .
44. And once we are done with it, we have another one,
45. which is an AM select function.
46. We do that.
47. So this is my five-step sequence, which is something
48. that you're already familiar with.
49. So what is different to make it an analog ADC pin
50. is the next sequence of steps.
51. So our first step that's specific to ADC is we turn ADC clock.
52. Like we turn on the clock here, we'll turn this one on.
53. We do that by making sure that RCGC0 bit 16 is set to 1.
54. So the next step we're going to do, which is step seven,
55. is we're going to configure the speed.
56. And the speed is the speed at which we're going to capture our samples.
57. There are multiple speeds ranging from 1 million samples per second all the way
58. to 125K samples per second.
59. And we effect that-- we make that change--
60. by making sure that we manipulate these two bits.
61. And for now, I'm going to make these two bits
62. 0 0, which says that I've chosen the 125K.
63. So that's 125K samples per second, which is
64. the minimum speed at which I can sample.
65. The eighth step, then, is our sequencer.
66. We have four sequencers.
67. We don't have to worry about them.
68. We're using sequencer 3.
69. Sequencer 3 is chosen.
70. And we also set the priority of it.
71. So because there are four sequencers, the priority is set by a two-bit number
72. here.
73. And we're making it the highest priority by making it 0 among the four.
74. The step nine is while I'm configuring the rest of the system,
75. I'm going to disable.
76. And when I'm done at the end, which is my last step, which is step 13,
77. I will enable.
78. This way, I'm making sure that while I'm configuring it,
79. I can make sure that the sampling doesn't occur.
80. So I do that by manipulating this bit, which is my activation bit.
81. So I turn this off.
82. And I turn it on to affect that.
83. So it goes from 0 to 1 eventually.
84. So our next step is step 10, which is we choose the trigger.
85. That is, when we have analog input connected,
86. it can be triggered by various mechanisms.
87. But in this class, we're going to use a software-initiated trigger.
88. So it means that I will not expect any interrupt or anything.
89. I'm simply going to tell when I want to choose to sample.
90. And when the sampling is done, I will read it myself.
91. So I do that by manipulating these four bits here.
92. And these four bits for us is going to be the simplest case, which
93. is all 0s, which says that I'm using software as a trigger.
94. PROF.
95. JONATHAN VALVANO: Professor Yerraballi, how do we
96. tell it which channel to sample?
97. PROF.
98. RAMESH YERRABALLI: OK, so everything we did here doesn't really
99. tell us anything about what channel we're using.
100. So our next step will involve exactly that,
101. which is my step 11, which says that I want to choose channel 1.
102. Remember, there are 12 channels.
103. They go from 0 through 11.
104. And so I'm going to choose channel 1 out of those.
105. So 1 is the I'm trying to choose.
106. So I need to specify that.
107. And I do that by writing a 0, 0 0 1 here.
108. And that's my channel select.
109. Channel is AIN1.
110. PROF.
111. JONATHAN VALVANO: Now is channel 1 always connected to PE2?
112. Or is that some kind of choice?
113. PROF.
114. RAMESH YERRABALLI: It is always connected to PE2.
115. And it's hard-wired.
116. And that's why we have selected the alternate function for PE2
117. to be its alternate function, which happens to be analog AIN1.
118. PROF.
119. JONATHAN VALVANO: Ah, OK.
120. One more step.
121. PROF.
122. RAMESH YERRABALLI: So we have one last step left, step 12,
123. which involves manipulating these four bits.
124. The only bit of real interest to us is this bit here, which is the IE0 bit.
125. And what it's telling us is that we want a flag to be set when the sampling is
126. complete-- set flag on sample capture.
127. The other bits, for now, are going to be 0 here, a 1 here, and a 0 here.
128. So that's going to be 0 1 1 0 into this, which
129. is a 6 is what we're going to write.
130. PROF.
131. JONATHAN VALVANO: So we do this once.
132. And so next, we're going to have to show you
133. how to write software that actually does the conversion-- starts, waits
134. for it to be done, and captures it.
135. PROF.
136. RAMESH YERRABALLI: That is correct.
137. So we looked at the initialization ritual.
138. Let's look at the sample capture procedure.
139. So to capture a sample, I'm going to follow this logic, which
140. says start with initialization after sampling by writing to this bit a 1.
141. So I make that bit a 1.
142. And then the ADC device starts its capturing.
143. And then I'm going to check for a flag.
144. So I'm going to check whether RIS bit 3 is a 0 to indicate that it is Busy.
145. So if it's busy, I keep checking back again and again.
146. And when it's 1, it tells me that the capture is complete.
147. So once the capture is complete, I'm going to read the data.
148. So I will read data, which means that I've done this busy step, which
149. I found this to be 0 for a good bit of time and eventually it became a 1.
150. So I come down here and then I'm going to read this data.
151. So read data.
152. So the last thing I need to do is to make sure that I clear the flag.
153. I clear this flag by writing a 1 here.
154. So the act of writing a 1 there will make sure
155. that this flag goes back to a 0 so I'm ready for the next sample.
156. And then I return.
157. PROF.
158. JONATHAN VALVANO: I have a question.
159. Do these registers act like memory?
160. In other words, if I write a 1 to them, it will become a 1?
161. And if I write a 0, it will become a 0?
162. PROF.
163. RAMESH YERRABALLI: Well, these are really not memory registers.
164. These are device registers.
165. PROF.
166. JONATHAN VALVANO: Oh.
167. PROF.
168. RAMESH YERRABALLI: So they don't behave like memory.
169. They behave more like inputs and outputs to a device.
170. PROF.
171. JONATHAN VALVANO: So if I write to one register,
172. it clears a bit in another register?
173. PROF.
174. RAMESH YERRABALLI: Yeah.
175. Because it's a memory-mapped I/O.
176. PROF.
177. JONATHAN VALVANO: I see.
178. Cool.
179. All right.
180. Enough talking.
181. Let's build it.
182. PROF.
183. RAMESH YERRABALLI: All right.