Introduction

Hello. My name is Alan Hawse. I'm Senior Vice President of Technical Staff for Solutions and Software here at Cypress Semiconductor. Welcome to Cypress Academy. This is the first video of a series of 40 or so videos that are going to provide you short, fundamental lessons in developing ultra-low-power, embedded IoT solutions using our PSoC 6 Microcontroller. PSoC 6 is Cypress’ solution for the next generation embedded IoT applications – your applications. With its dual core ARM Cortex-M4 and M0+ architecture, it delivers the performance that you need for your applications; it delivers security features that are critical for your products not getting hacked – this is true for all connected devices and it's something you should be particularly aware of. This device also enables innovative solutions because it's highly flexible and it's programmable - while all of this is built on an ultra-low-power MCU architecture. But I’m not really in marketing - I’m an engineer…so let me show you so you can believe what PSoC 6 can do.

In this video series, I’m going to take you through a complete example all the way from soup to nuts for building an IoT application. In this video I'll show you what I'm going to build, and then in the rest of the videos that go with this series, I will take you through each of the building blocks – UART, I2C, CapSense, BLE – and I'll show you all of the steps required to realize this vision.

So, what are we going to build? Well, my son Nicholas – he really loves robots. So, we’re going to build a BLE-controlled robot arm using this Amazon.com robot that I got. It's not going to just be any old robot…it is going to be like one of those rise of the machine robots. It's going to do all of the terminator stuff. It's going to have kill switches; it's going to have BLE; there's going to be a wireless remote control; it's going to have RTOS’; it's going to have capacitive sensing; there's accelerometers; it's going to have thermistors, and RTC. I'm going to show you how to do trace analyzers and do RTOS debugging. I'm going to teach you secure bootloading to make sure your product is safe. And then, when we get to the end, I'm going to teach you how to make it on WiFi as well and probably, I think I'm going to throw in some speech to text processing using web services.

That's going to be really awesome but you might ask yourself "Why build something so complicated? The real answer is when Nicholas watches the videos, I don't want him to say it's boooooooring… but really why do I do this? Because I think it’s going to help you when you develop your next generation IoT application…because each step of the way I’m going to show you how the building blocks work and how they fit together. The whole thing is going to be messy - probably really messy - we’re both going to learn things together as we go through the process but in the end, I think we are going to build something awesome.

As we go, you might have questions, or you might get stuck somewhere, so through each step of the process there will be references to our website. It will have application notes, and code examples, and other references to places on the internet. Also, if you have any questions along the way, you should post them in the PSoC 6 community on Cypress.com. Or please feel free to email me at [alan\_hawse@cypress.com](mailto:alan_hawse@cypress.com) or tweet me @askiotexpert.

So, let’s get started…

For this class I will build all of these projects on this development kit, called the PSoC 6 BLE Pioneer Kit, also known as the CY8CKIT-062-BLE. This kit ships with three boards in the box: an Arduino-Uno compatible base board - this is the main board that has the PSoC 6 BLE MCU on it; it also ships with an E-ink display shield that has some sensors on it; and a CySmart BLE 4.2 USB dongle. The PSoC 6 BLE base board has a lot of stuff on it:

* PSoC 6 BLE MCU with 1MB of internal flash. It has dual ARM cores – and ARM Cortex-M4 as well as an ARM Cortex-M0+
* A PCB antenna for the BLE radio
* PSoC 5LP MCU that's used for debugging and programming called KitProg
* An external 512Mb Cypress NOR flash IC that you can save your stuff on
* Arduino-Uno compatible headers for using other Arduino shields
* EZ-PD CCG3 USB Type-C power delivery system
* An interface area for CapSense sliders and buttons
* RGB LED

The E-ink display shield has:

* E-ink display
* PDM microphone
* Thermistor
* 9-axes motion sensor including magnetometer, gyroscope, and accelerometer

In addition, I’m using a basic Arduino shield. It's just there to connect the wires of the PWM up to the servo motors. I'm going to create a custom version of the board later and you'll be able to find details – like the board design files, etc. – on my blog at [www.iotexpert.com](http://www.iotexpert.com). I'll also add a link to that content next to where you're watching this video.

The robotic arm I'm using has 4 servo motors. Each of the servos require power, ground, and the pulse width modulator (PWM) input. The breakout board is optional, but it helps simplify the wiring of the PSoC 6 BLE Pioneer board to the robotic arm. These connectors have 3 standard wires – the power, the ground, and the PWM signal – and it makes it simpler to connect up with my breakout board.

Through this video series we might use some additional hardware, for example a Cypress WiFi module or a second PSoC 6 BLE Pioneer Kit. I’ll introduce those pieces when we get there.

The class has 4 chapters which I will turn into the 40ish videos that I'm talking about. They are:

1. System Architecture (Design flows, FreeRTOS, etc.)
2. Using the MCU peripherals (GPIOs, PWMs, ADCs, Smart IOs, etc.)
3. Building a BLE remote control (BLE, CapSense, E-ink display, motion sensor, etc.)
4. Connecting it all to the Internet with Wi-Fi and cloud services

Again, I would encourage you to post your comments and questions in the PSoC 6 developer community on Cypress.com. Or, as always, feel free to email me at alan\_hawse@cypress.com or tweet me at @askioexpert and please send me your comments, your suggestions, your criticisms and your questions.

Alright, let's go make an amazing IoT project.