Build Arduino Apps Like a Pro
With the Arduino C++ IDE for Eclipse

Doug Schaefer, BlackBerry/QNX
Eclipse CDT Co-Lead
Outline

- What is Arduino
- Why Eclipse for Arduino
- Introducing Arduino C++ IDE for Eclipse
- Future Plans
What is Arduino?
What is Arduino?

- Open hardware prototyping platform with an open source SDK and IDE
- Multiple microcontroller architectures but with common API
  - Atmel AVR 8-bit
  - ARM Cortex-M0+ 32-bit
  - ESP8266's Tensilica L106 32-bit
- Very limited computing resources
  - E.g. Uno: 16 MHz 8-bit CPU, 32KB flash, 2K SRAM, 1K EEPROM
  - ESP8266 – 80MHz, 64/96KB I/D RAM, up to 4MB flash
- Interface to outside world using GPIO, USB Serial, SPI, I2C
  - SDK makes it really easy to program these interfaces
The Arduino IDE

```cpp
#define PIN 7

void setup() {
    pinMode(PIN, OUTPUT);
    Serial.begin(115200);
}

void loop() {
    Serial.println(F("Hi"));
    digitalWrite(PIN, HIGH);
    delay(500);
    digitalWrite(PIN, LOW);
    delay(500);
}
```

Done compiling.

Sketch uses 2,442 bytes (7%) of program storage space. Maximum is 32,288 bytes (99%). Global variables use 200 bytes (9%) of dynamic memory, leaving 1,848
The Arduino IDE

- Great for beginners.
  - Lots of tutorials on how to get started

- Does everything you need.
  - Build, upload, serial port console
  - Library and Board SDK management

- But very simple.
  - “Arduinopad”
  - No content assist, search
  - No source control integration
  - No debugging*
Why Eclipse for Arduino?

- For Arduino programmers that want a full featured IDE
  - With content assist, search
  - With source control integration
  - With hope of having debugging*

- Access to the Eclipse IDE ecosystem of plug-ins
  - Arduino as a component in a larger system, e.g. IoT device to cloud
  - Potential for cross-platform search, code generation, “I” in Integrated

- Because I bought an Arduino and want to use Eclipse for it 😊
Exemplary Implementation of Eclipse IDE Frameworks

- New CDT Build System
  - Minimalistic but with support for CDT source navigation features
  - Not based on managed build
  - Uses same config files as Arduino IDE

- Launch Bar
  - Arduino devices as Launch Targets
  - One click build and launch

- Remote Target Management System
  - Extends Serial Port connection type from org.eclipse.remote

- IDE Commons Code Template Framework
  - For generating projects with initial boiler plate code
Current Status

- Preview Release based on Eclipse Mars October 2015
  - Up on the Eclipse Marketplace
  - 2600 downloads to date
- How to video on YouTube
  - [https://www.youtube.com/watch?v=CPqsJUXFFnQ](https://www.youtube.com/watch?v=CPqsJUXFFnQ)
  - 3900 views to date
  - Also did a webinar for the Eclipse Foundation
- Great feedback from the Community
  - 65 bug reports and enhancement request in Bugzilla, 30 still open
  - Great questions in CDT forum, YouTube comments, Twitter
Demo
Future Plans

- Production quality release for Eclipse Neon in June
- Migrate to final APIs for build and launch
- Improved support for Libraries
  - Number one complaint and requested feature
  - Both imported via zip file and Arduino Library projects
- USB device detection
  - To auto-populate targets based on vid/pid
- Fix as many bugs as I can
  - Hopefully review and apply patches from our helpful community
Future Directions

- Leverage Arduino config files for other SDK’s
  - E.g. FreeRTOS for ESP8266 and ESP32
- Support import of Arduino sketches
  - Editor and build support for INO files
- Investigate debugging
  - Gdb stub linked with application?
  - JTAG integration via CDT’s gdb JTAG support
- Static Code Analysis
  - CDT Codan checkers to detect size/performance issues
Thank you!