A Quick Look at Several Microcontroller Boards

Dave Blevins

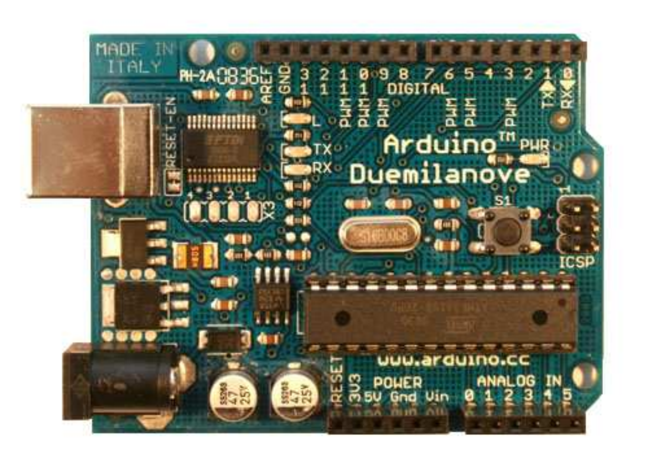
October 2014

Arduino

- http://arduino.cc/
- Open Source hardware and software (IDE)
- Most versions based on 8-bit Atmel ATmega processor (but newest versions use an ARM-type chip)
- The Duemilanove version uses an ATmega328p
- Typically runs at 16mHz (some low power versions run at 8mHz or lower)
- Older versions run at 5V; newer ones are 3V3
- Relatively robust I/O pins
- Typical board starts at about $30

Duemilanove

("2009" in Italian)



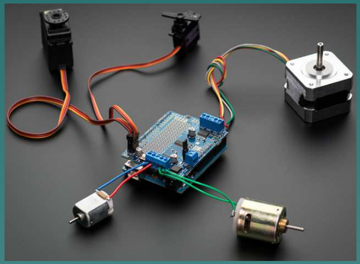
- 14 digital I/O pins (6 can do 8-bit Pulse Width Modulation)
- Six 10 bit (1024 value) analog inputs
- Memory: 32KB Flash (for programs), 1KB SRAM, 1KB EEPROM
- Powered either via USB or 7->20VDC input jack
- Programmed via USB
- Has both 5V and 3V3 power outputs
- Can interface with TTL serial, SPI and i2C devices
- Programmable pullups on digital I/O pins

Accessory Boards: “Shields”

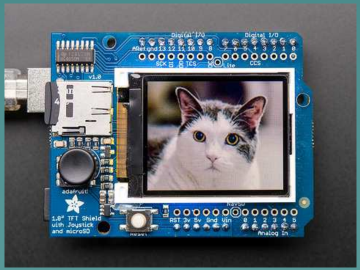
- Shields plug into top of Ard board – form factor is standardized
- In some cases, shields can be stacked

Lantronix X-Pico Wifi Shield

- Full TCP/IP stack
- Web application server
- Serial or SPI interface to Ard

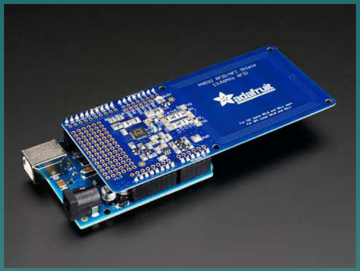


Motor/Stepper/Servo Shield

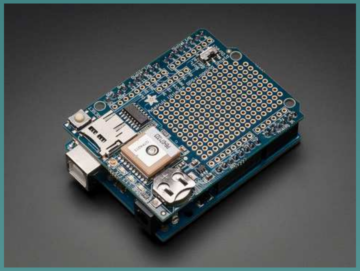


1.8” Color TFT Display

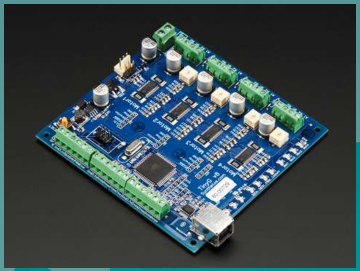
- SD card slot
- Tiny joystick



NFC/RFID Controller

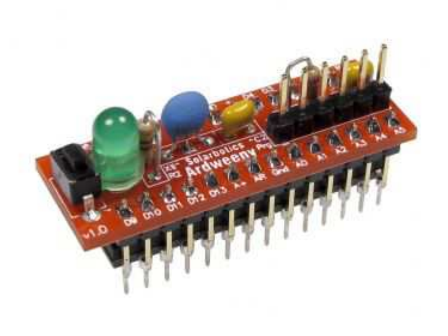


GPS Logger



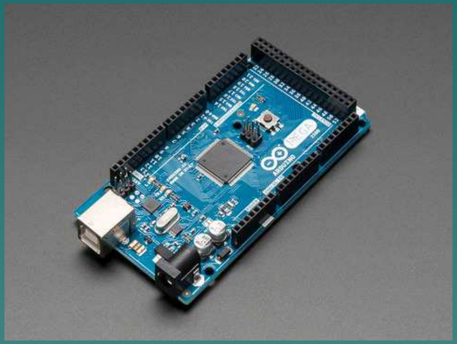
TinyG CNC Controller

Some other form factors



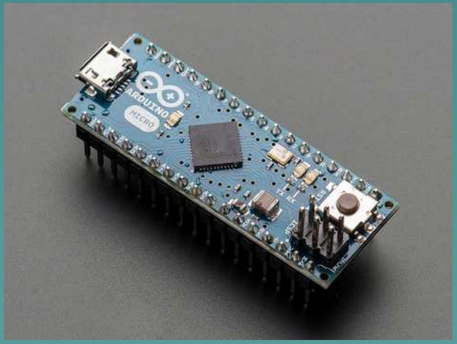
Ardweeny

Robot

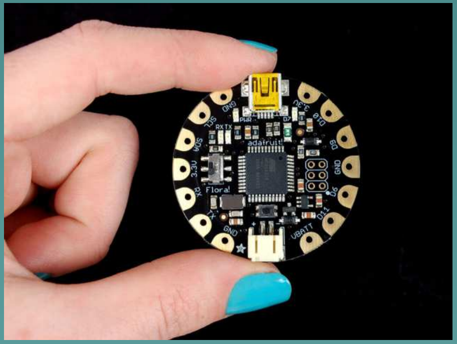


Mega

- ATmega2560
- More I/O pins

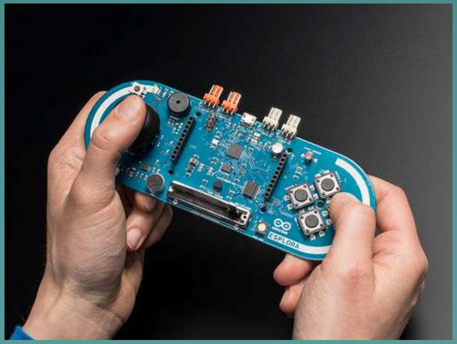


Micro



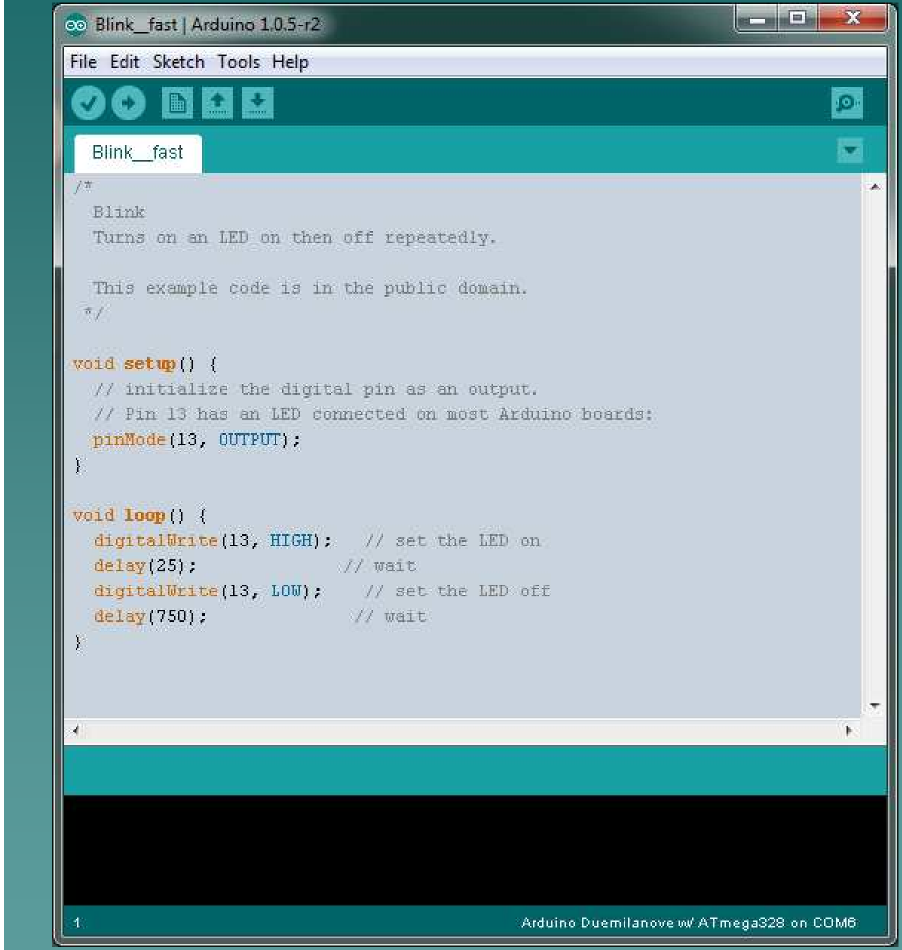
Flora

(for wearables)



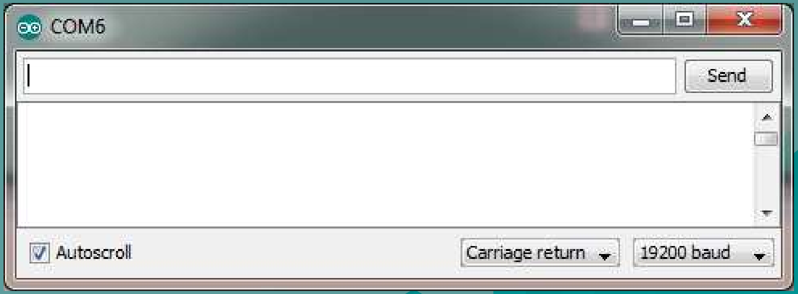
Esplora

IDE: “Arduino”


Blink_fast | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Blink_fast
1
Arduino Duemilanove w/ ATmega328 on COM6

```
/*
  Blink
  Turns on an LED on then off repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(25);              // wait
  digitalWrite(13, LOW);    // set the LED off
  delay(750);              // wait
}
```

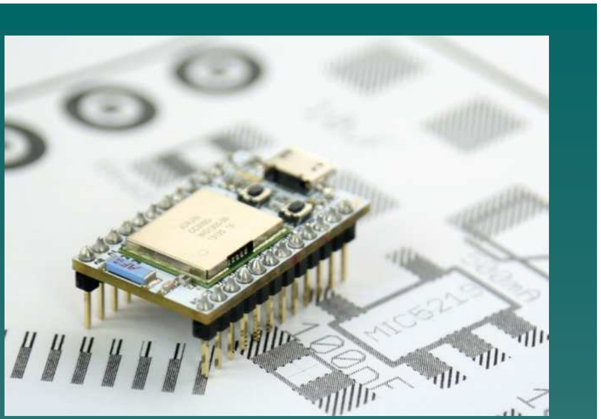
- OS X, Linux, Windows versions
- Language is similar to C++
- Programs are called Sketches
- setup() and loop() sections
- loop() runs “forever”
- Serial monitor window assists in code debugging and allows “talking” to the Ard via keyboard



☑ Autoscroll Carriage return 19200 baud

Spark Core

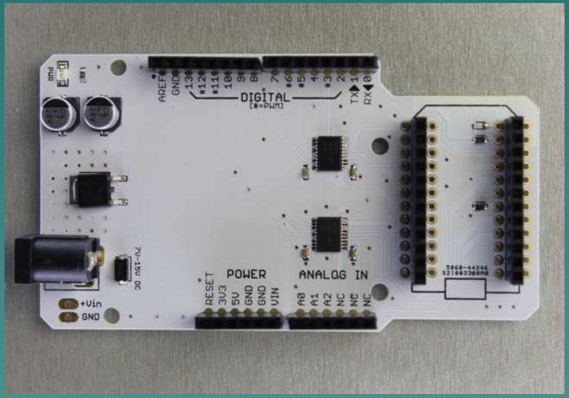
(Started out as a Kickstarter project)

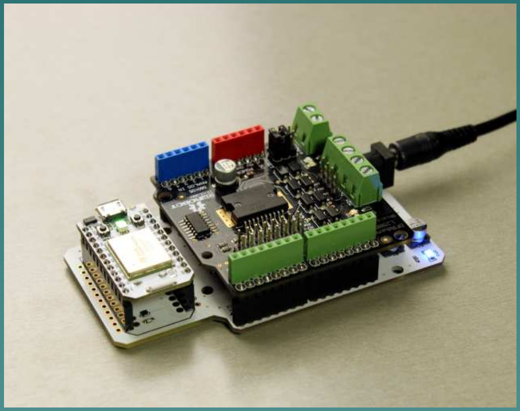


- Open Source
- Arduino language compatible – up to a point (in some cases, peripheral support libraries have to be rewritten)
- ARM Cortex 3 microprocessor
- 32 bit, 72mHz
- 2MB (external) & 128kB(on-chip) Flash, 20KB SRAM
- 8 digital, 8 analog pins
- SPI, i2C, TTL serial, JTAG
- TI CC3000 wireless chip: 802.11b/g
- 100-300 ft range
- Power: 3V3, 100-300mA
- $35

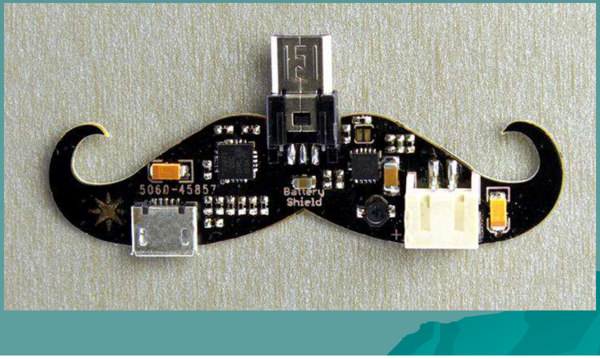
Spark Core: Shield converter

- Allows Arduino shields to be used with the Core
- Digital pins are level-converted from the Core’s 3V3 to 5V





- A few other Core-specific shields are available – e.g. the Battery Shield!



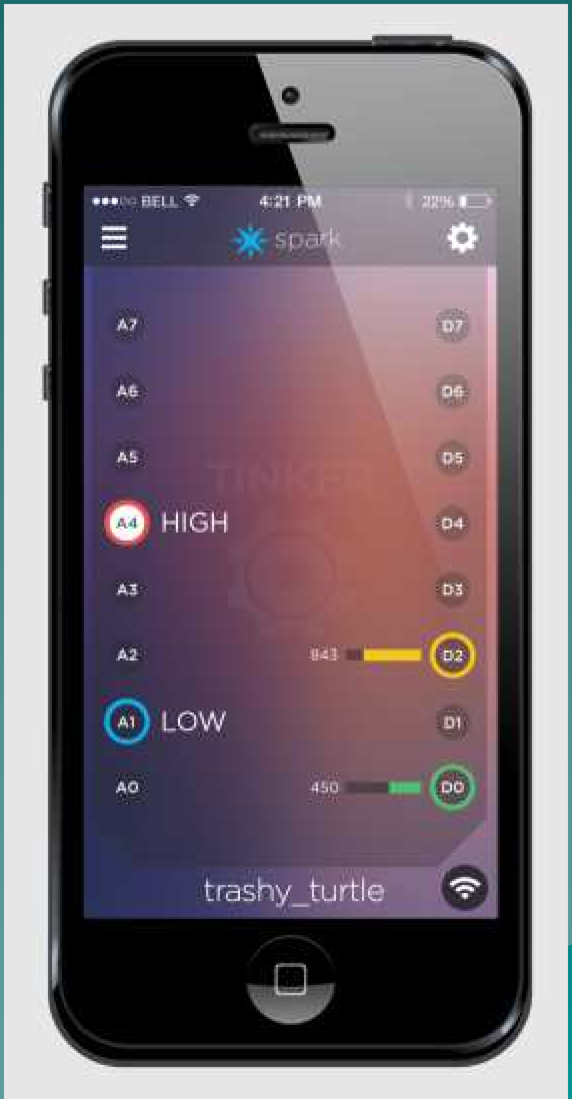
Spark Core: IoT

- REST API
  - **RE**presentational **S**tate **T**ransfer
  - Basically, this means you can communicate with one or more Spark Cores via HTTP, once the Core is connected to the Spark Cloud
  - Every Core has a URL
  - Security: authentication through OAuth2 (each Core has a unique Access Token)

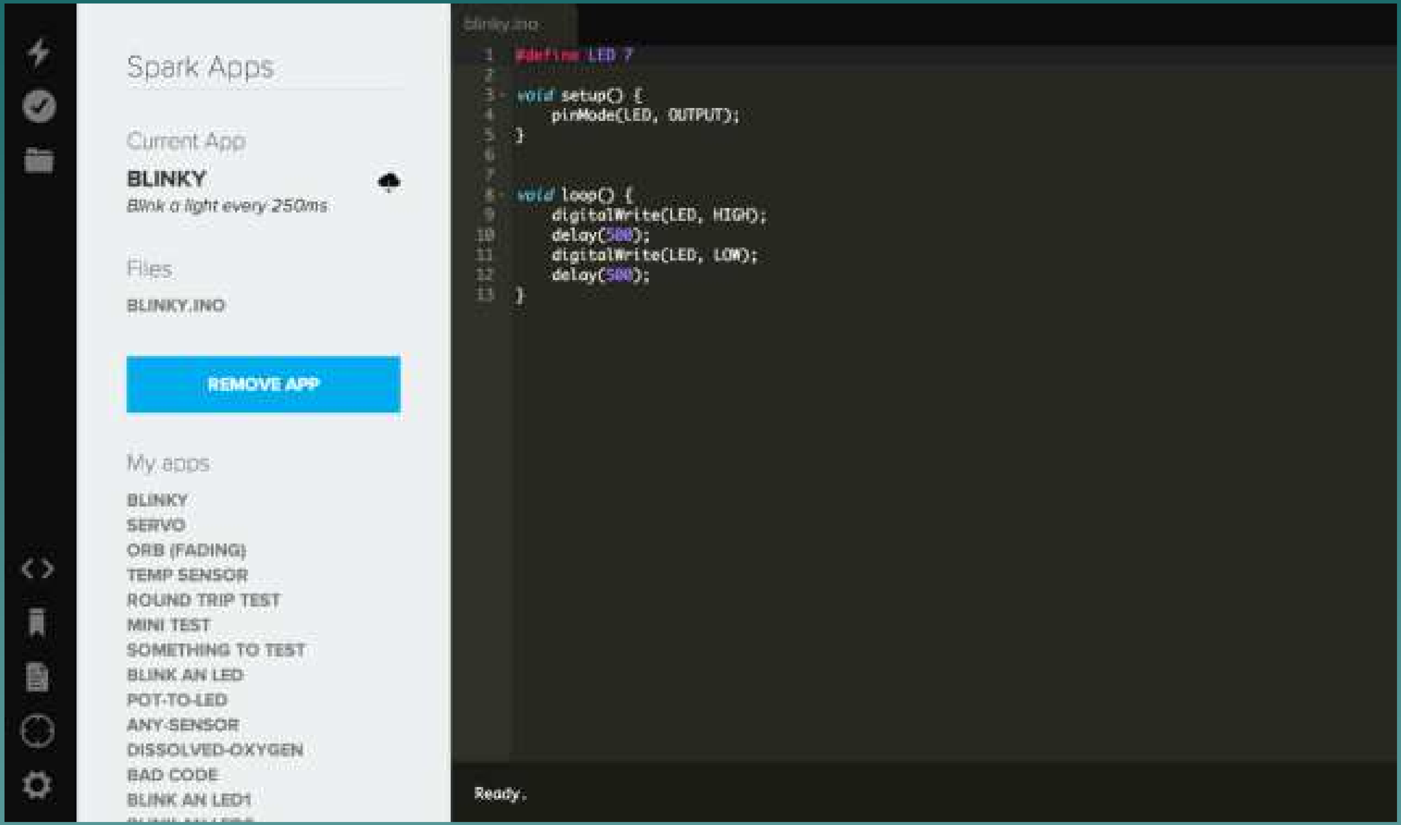
```
Query:
curl "https://api.spark.io/v1/devices/temperature?access_token=12341234123412341234"

The response contains a result like this:
{
        "cmd": "VarReturn",
        "name": "temperature",
        "result": 42,
        "coreInfo":
        {
                "last_app": "",
                "last_heard": "2014-08-22T22:33:25.407Z",
                "connected": true,
                "deviceID": "53ff6c065075535119511687“
        }
}
```

The Tinker App

- A basic demonstration app that shows how a Core can be controlled/monitored from a mobile device
- Core pins can be configured to be analog or digital inputs or outputs
  - Note that Spark Core's analog output is actually digital PWM



Spark Core IDE



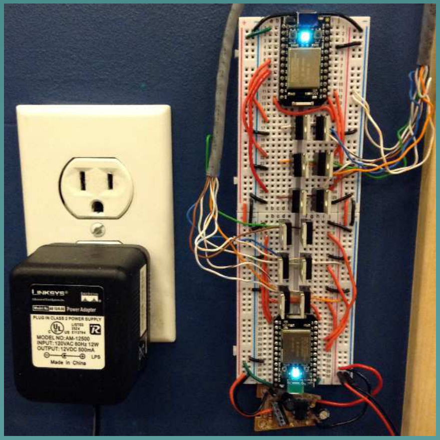
```
#define LED 7

void setup() {
    pinMode(LED, OUTPUT);
}

void loop() {
    digitalWrite(LED, HIGH);
    delay(500);
    digitalWrite(LED, LOW);
    delay(500);
}
```

```
Ready.
```

- Web-based, but uses Arduino language
- Spark Core programmed wirelessly via the “Spark Cloud”

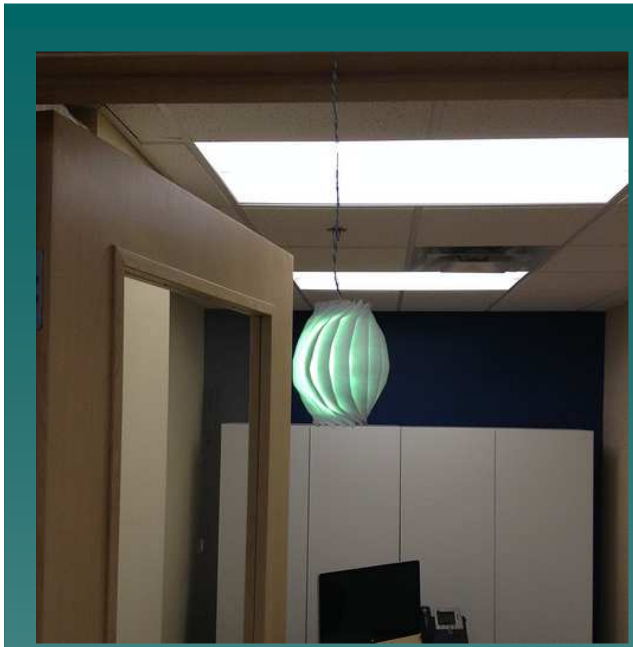
Example Spark Core Application: Illumigami

- Basic Idea: use Cores to control colored lights that indicate the status of meeting rooms:
  - Green - Available
  - Yellow - Available now, Booked in the next 10 minutes
  - Red - Booked
  - Blue - Booked now, Available in the next 5 minutes
- Uses REST API to send Google Calendar data (via GC's API) to lights

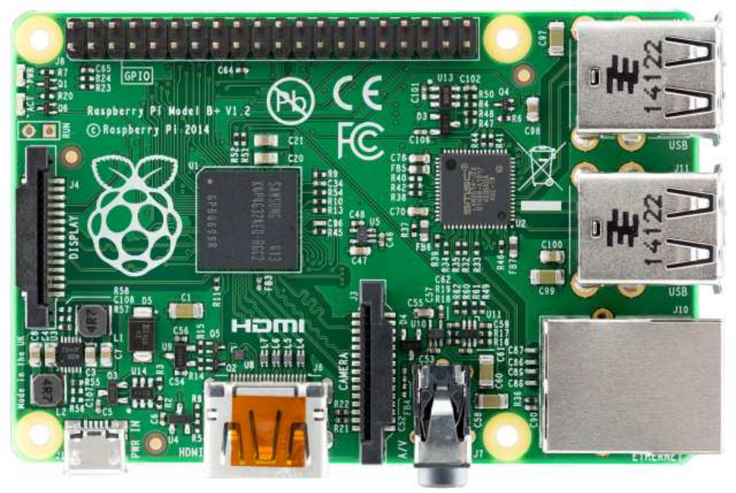


RGB LED strip used for lighting

http://viget.com/extend/illumigami-the-hero-conference-rooms-deserve



Raspberry Pi



- Much, much higher performance than an Arduino-class device (of the Atmel ATmega variety)
- Runs an actual OS
- Started out as an educational tool to be used in schools in England to teach programming
- "Pi" refers to the primary programming language, Python
- Uses a Broadcom SoC chip, which includes an ARM6 core as well as a "VideoCore" for graphics
- B model is $35; A model (no Ethernet) is $25

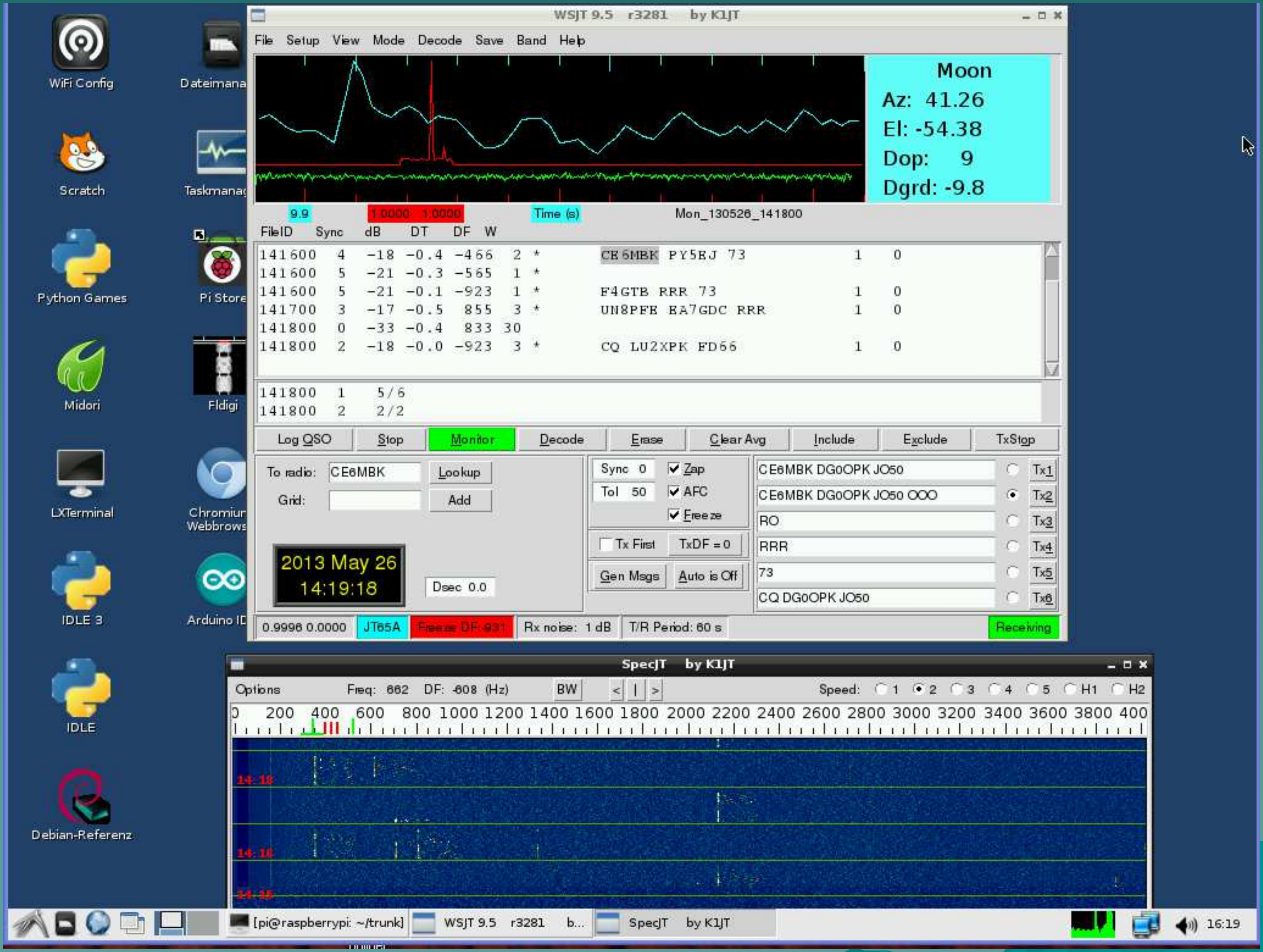
Pi Bits ("B" model)

- 700mHz, 512MB RAM
  - Performance similar to a 300mHz Pentium II
  - Graphics performance on par with Xbox I (i.e. you can run Minecraft on it!)
- Boots OS from SD card
- OSs: Debian, Arch, FreeBSD, NetBSD, OpenWRT (router OS), etc.
- HDMI, composite video outputs
- Analog audio out (in addition to digital audio of HDMI)
- Ethernet port
- 2 USB ports (for keyboard, mouse, …)
- GPIO connector (3V3) supports i2C, SPI, TTL serial, PWM
- Powered via power-only (no data) micro USB port
- Connector provided for dedicated camera

More Pi for you?

- Raspbian Linux distro has a windowed GUI
- Can be run “headless” (i.e. via remote desktop)
- No built-in wireless, but a wireless USB dongle can be used
- GPIO pins are not as robust as the Ard’s and should be buffered to protect against board damage by accidentally hooking something up wrong
- Some uses for a Pi: XBMC media center, open source router, web server, file server…
- Latest “B+” model adds more GPIO pins, 2 more USB ports, and a number of improvements

Raspbian GUI



WSJT 9.5 r3281 by K1JT

Moon
Az: 41.26
El: -54.38
Dop: 9
Dgrd: -9.8

9.9 1.0000 1.0000 Time (s) Mon_130526_141800

```
FileID   Sync  dB   DT    DF  W
141600   4    -18  -0.4  -466  2 *      CE6MBK PY5EJ 73          1   0
141600   5    -21  -0.3  -565  1 *
141600   5    -21  -0.1  -923  1 *      F4GTB RRR 73             1   0
141700   3    -17  -0.5   855  3 *      UN8PFE EA7GDC RRR        1   0
141800   0    -33  -0.4   833 30
141800   2    -18  -0.0  -923  3 *      CQ LU2XPK FD66           1   0

141800   1    5/6
141800   2    2/2
```

To radio: CE6MBK
Grid:

2013 May 26
14:19:18

Dsec 0.0

Sync 0 ☑ Zap
Tol 50 ☑ AFC
☑ Freeze
☐ Tx First TxDF = 0
Gen Msgs Auto is Off

CE6MBK DG0OPK JO50	○	Tx1
CE6MBK DG0OPK JO50 OOO	●	Tx2
RO	○	Tx3
RRR	○	Tx4
73	○	Tx5
CQ DG0OPK JO50	○	Tx6

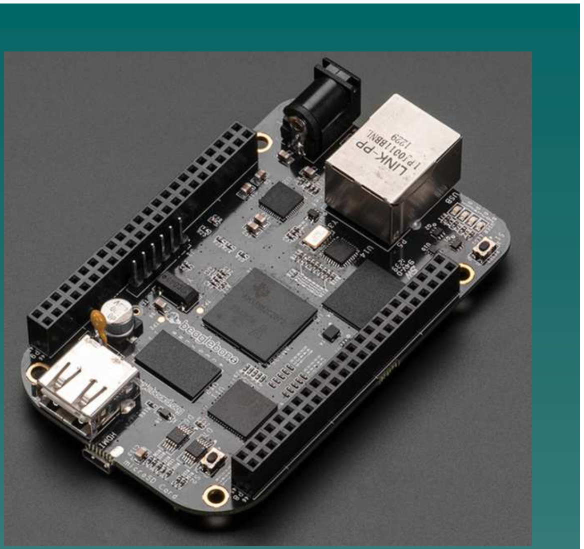
0.9998 0.0000 JT65A Freeze DF -931 Rx noise: 1 dB T/R Period: 60 s Receiving

SpecJT by K1JT

Options Freq: 862 DF: -608 (Hz) BW

Speed: ○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ H1 ○ H2

BeagleBone Black



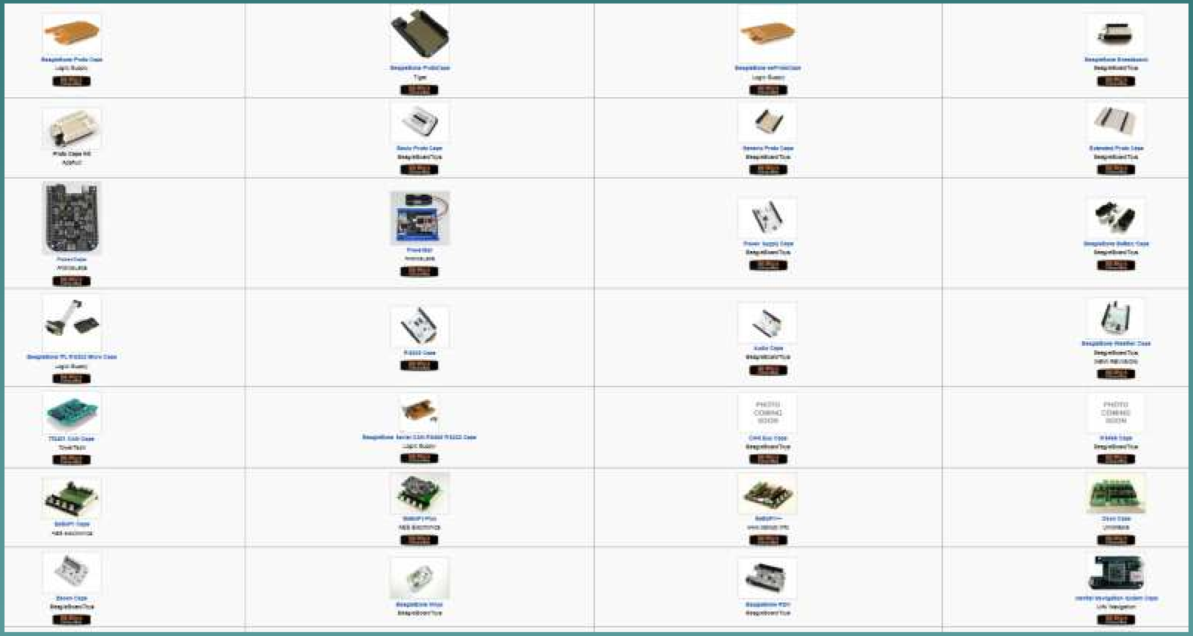
- Somewhat more powerful than the R-Pi
- Designed by Texas Instruments
- ARM Cortex A8 processor architecture
- Rather than Shields, BBB uses “Capes”
- Like the R-Pi, it runs various flavors of Linux (and also Android)
- $45

BBB

- Processor: TI Sitara AM3358AZCZ100
- 1GHz
- 4GB on-board flash storage
- SDRAM: 512MB DDR3L 800MHZ
- 3D graphics accelerator
- Floating-point accelerator
- USB host, Ethernet, HDMI
- SD/MMC Connector for microSD
- 2x 46 pin headers

BBB

- A large number of “capes” are available… Here’s a small sample:

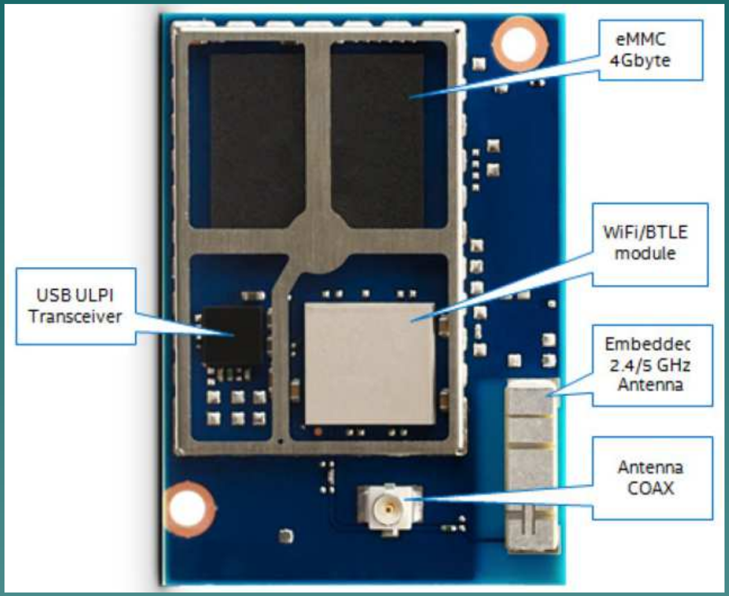


http://elinux.org/Beagleboard:BeagleBone_Capes

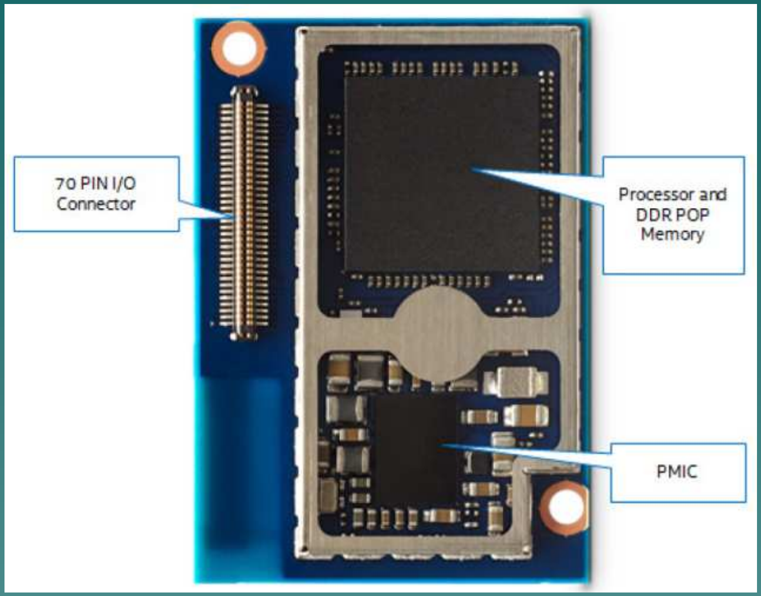
Intel Edison

- A new offering, this module is not really comparable to what we've looked at so far
- It's intended to be an embedded IoT controller
- Native voltage is 1.8V
- Main SoC chip combines a dual-core 500mHz Atom processor and a Quark 100mHz microcontroller (not currently enabled)
- The Atom core runs the "Yocto" Linux distro
- The Quark core will eventually run a "ViperOS" RTOS, which is derived from WindRiver's VxWorks
- On-board 802.11a/g/b/n and Bluetooth 4.0 (LE later)
- $50

Edison Layout & Specs



Top View



Bottom View

- 4GB eMMC*, 1GB LPDDR3 PoP** RAM
- 70 pin I/O connector (very very fine pitch), which includes USB, SD, UART, i2C, SPI, I2S, 12 x GPIO, and 32kHz / 19.2mHz clocks
- Power = 3V3 to 4V5 DC; current requirements not yet published for all operating modes

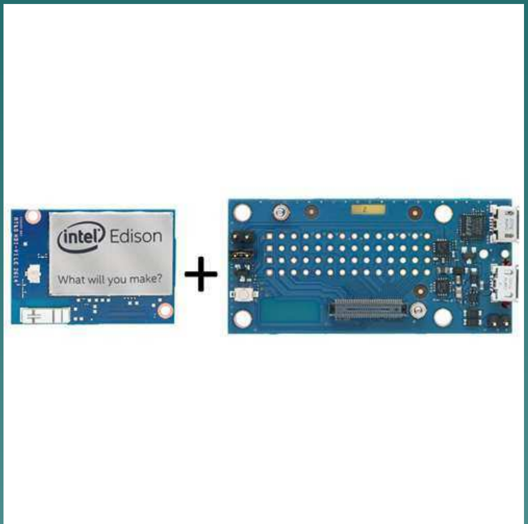
**eMMC = embedded Multi-Media Controller

**PoP = Package on Package

Edison Board for Arduino

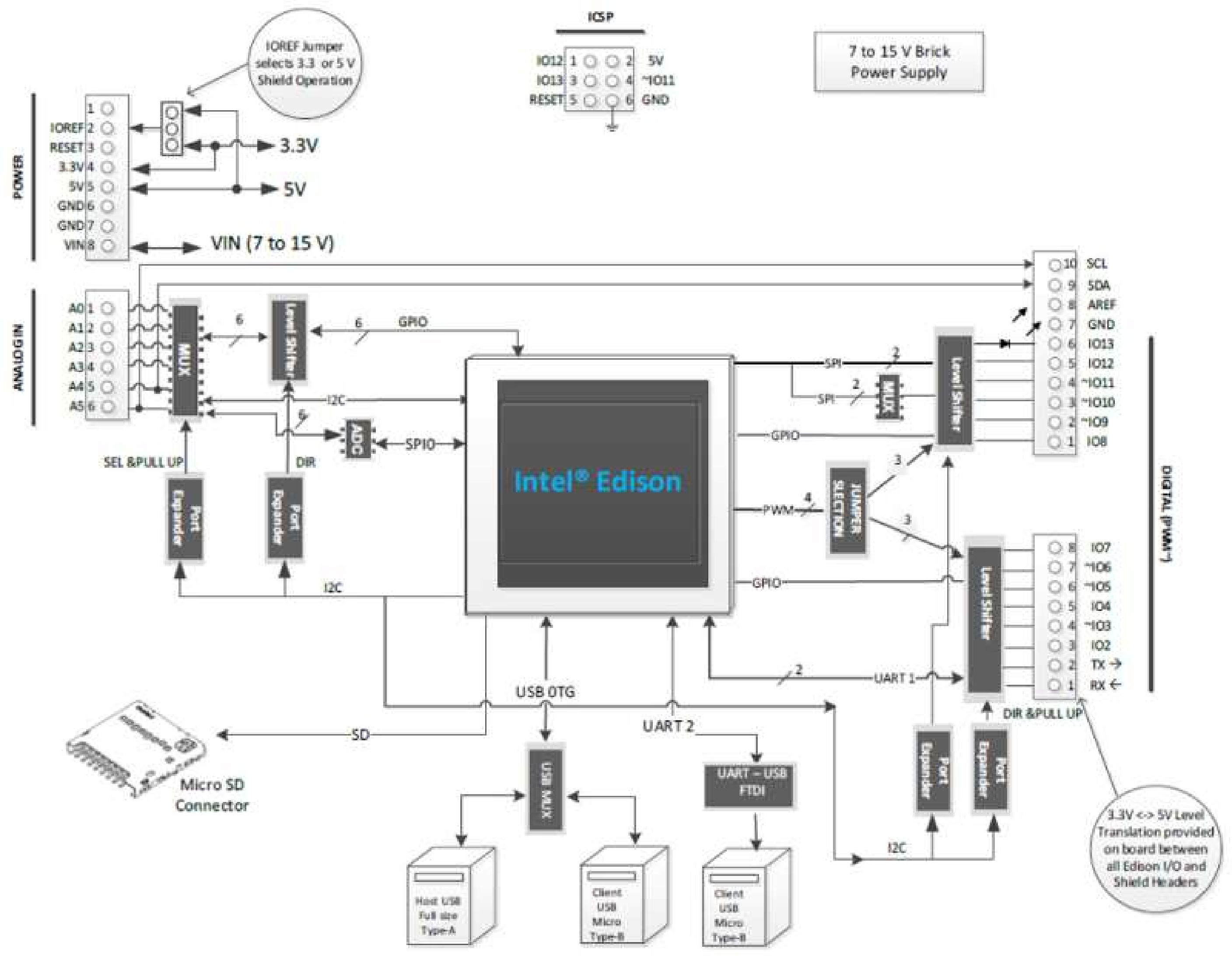
- Provides Arduino shield-compatible connectors, 1.8V to/from 3v3 or 5v level shifting, and Arduino I/O compatibility
- Edison/EBfA combo is $85 on Amazon

Edison Breakout Board; IDEs



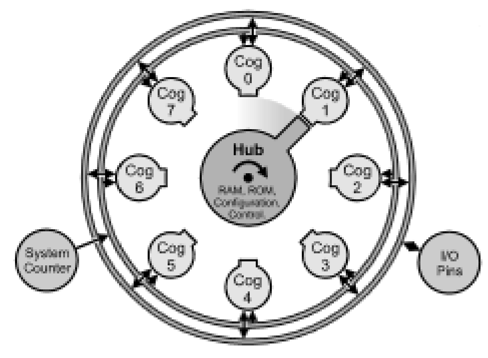
- Another, smaller development board option
  - 1V8 I/O
  - 0.1” grid of through-hole I/O access
  - USB interface
  - Charger circuit for LiIon battery
  - Power supply jack (7 – 15 VDC)
- IDEs
  - Customized Arduino IDE (Arduino emulation / functionality runs as a process under Linux)
  - Eclipse
  - Intel XDK IoT Edition

Appendix: Edison Board for Arduino Block Diagram



Parallax Propeller

- Multi-core: eight 32 bit RISC CPU “cogs”
- Verilog code for the Propeller is open source as of August 2014
- 2KB memory per core plus shared 32K RAM & 32K ROM
- 32kHz – 80mHz clock speed
- 32 I/O pins
- 160 MIPS (8 cores combined)
- Unused cogs can be shut down to conserve power
- No floating point hardware
- No explicit interrupt hardware – instead, a core is dedicated to waiting on an interrupt event to process



Hub and Cog Interaction

Propeller Code Development

- Two main languages
  - Spin: high level
  - Propeller Assembly
- Other languages
  - C, C++
  - Pascal
  - Forth
  - Java (under development)