EPICS Training (Hands-on)

KEK, High Energy Accelerator Research Organization
Takashi OBINA

Sakura Program
2018/01/30

Agenda

1. Visit PF Control Room

2. Training / Exercise / Hands-on

Purpose of the training

Main Purpose of Today

- Get familiar with EPICS
- Acquire a feeling of "Interaction between Hardware and Program"
- I hope you continue to enjoy after you go back to your institute.

In order to achieve the goal:

- Use cheap hardware : Raspberry Pi (RPi) or BeagleBone Black (BBB)
  - Both are 5,000 JPY (~45 USD, ~ 300 RMB)
  - Easy-to-use Digital I/O pin available
  - Analog Input (BBB)
- Learn about
  - Usage of DIO
  - Distributed control system
  - GUI : Control System Studio (CSS) or MEDM
  - Analog monitor

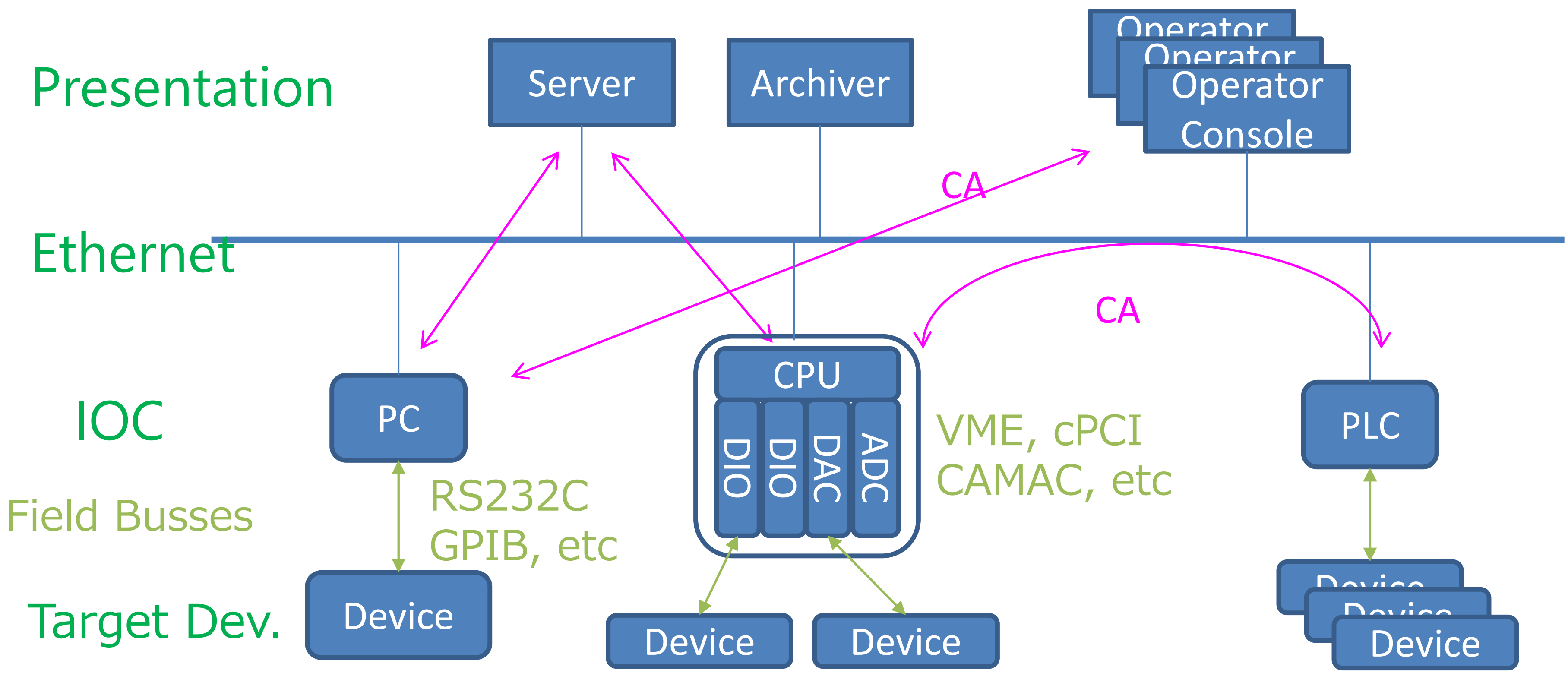
Procedure

- Introduction
  - What is Control Framework? Why do we need?
  - What is EPICS?
- Training Setup : Hardware
  - BeagleBone Black (IOC) + Solderless Breadboard
  - Device : LED, Switch, Temperature Sensor
  - Network (Ethernet)
  - Client PC (your laptop) + OPI (CSS or medm)
- Additional Topics (if time permits)
  - Remote Device (Digital Multimeter via Ethernet)
  - Different type sensors
    - Magnetic Sensor (analog)
    - i2c device, SPI device

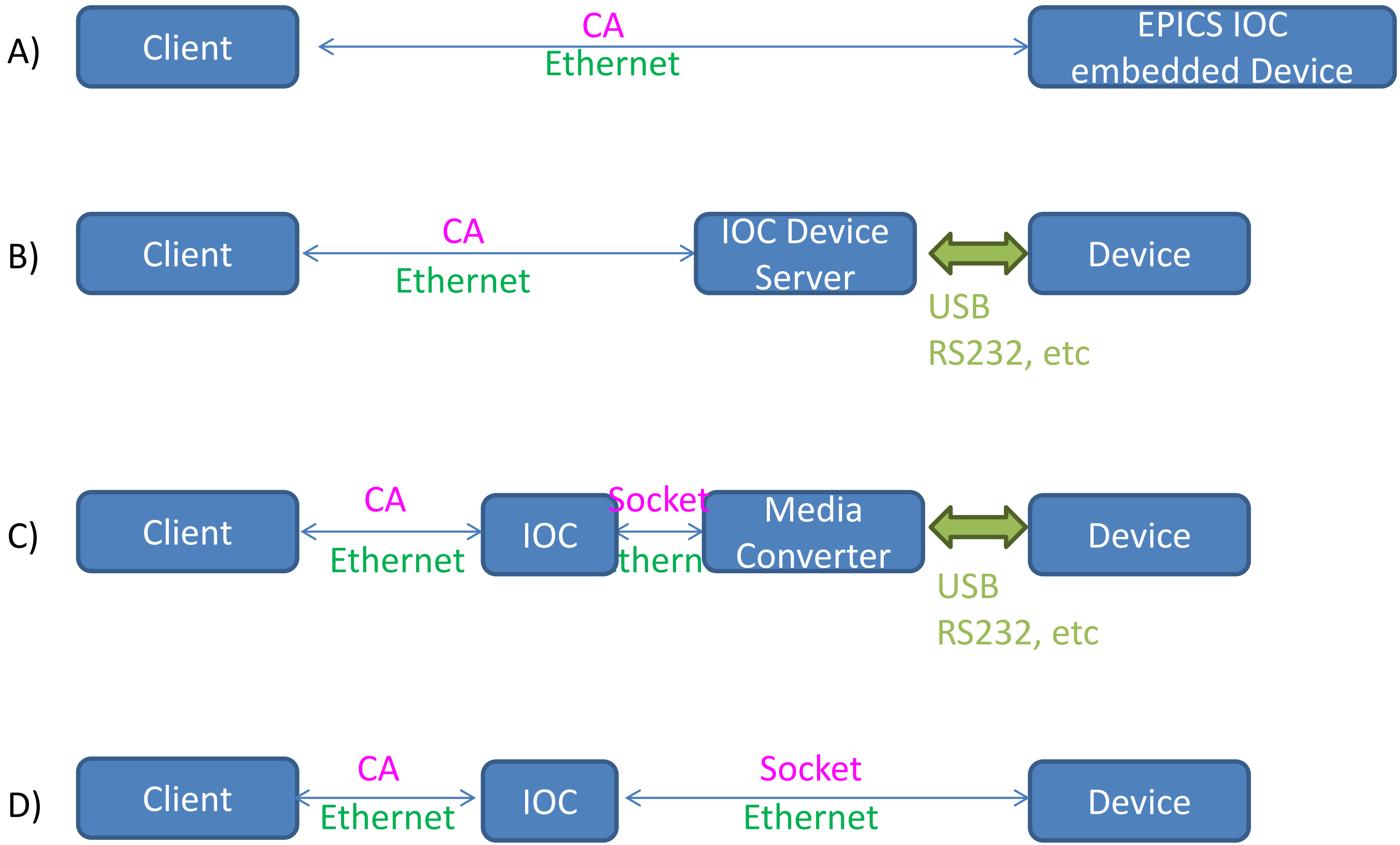
EPICS : Communication Protocol

Communicate with <u>Channel Access</u> protocol

- PV (Process Variable) in IOC
- Network transparent. Distributed system.
- CA protocol is also used for communication between IOCs

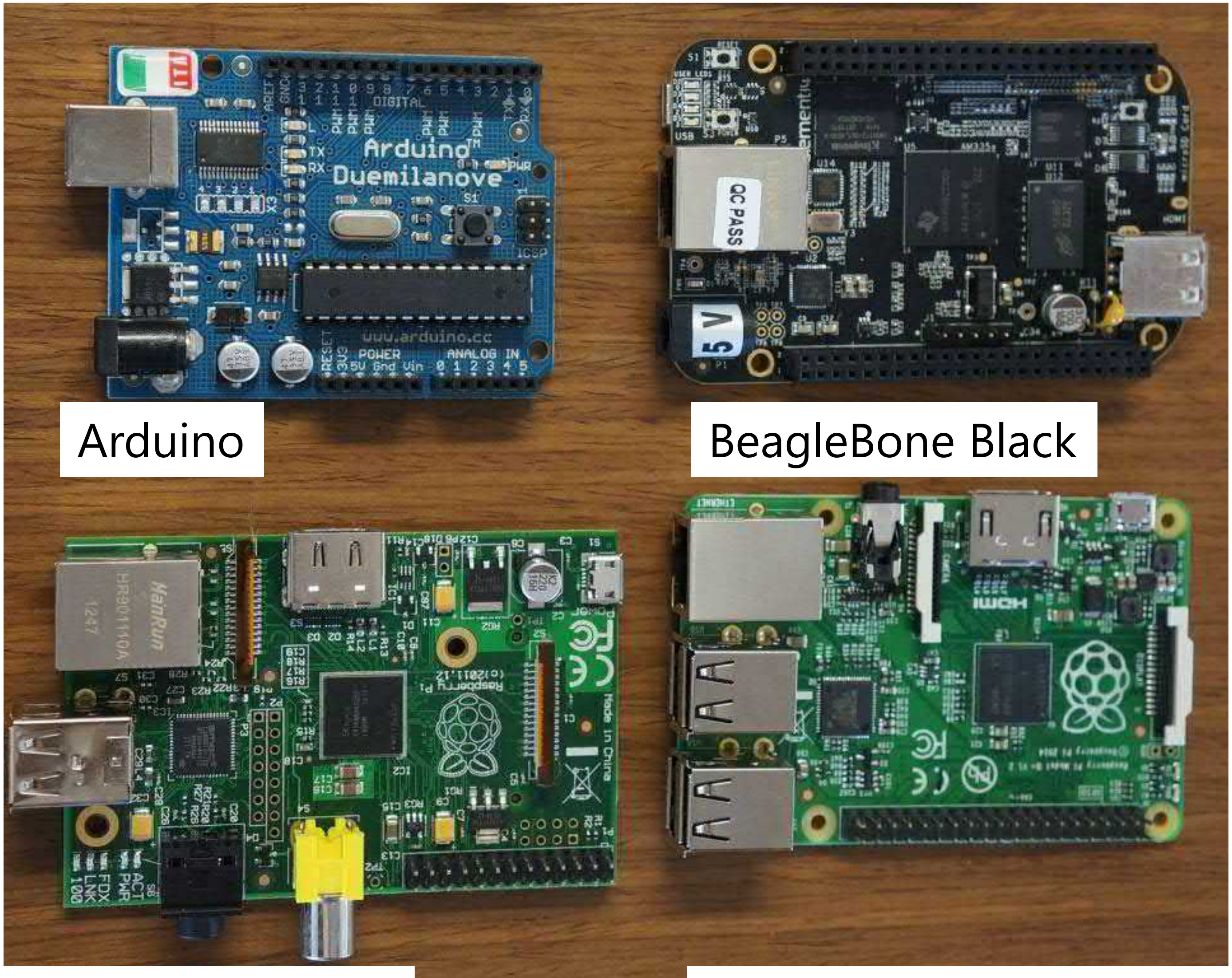


Comparison of configuration



Candidates

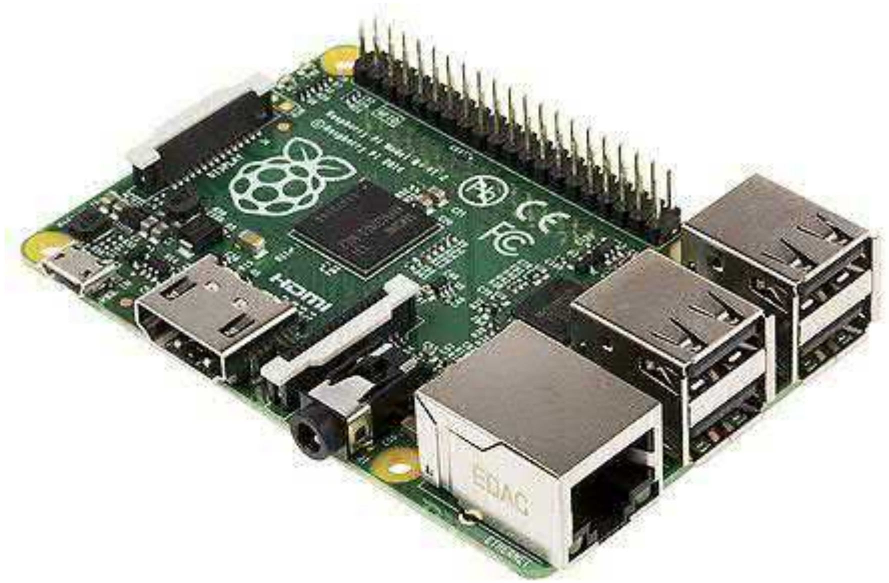
Credit-Card size board



Raspberry Pi (model B) Raspberry Pi (model B+)

Why BBB? Not Raspberry Pi?

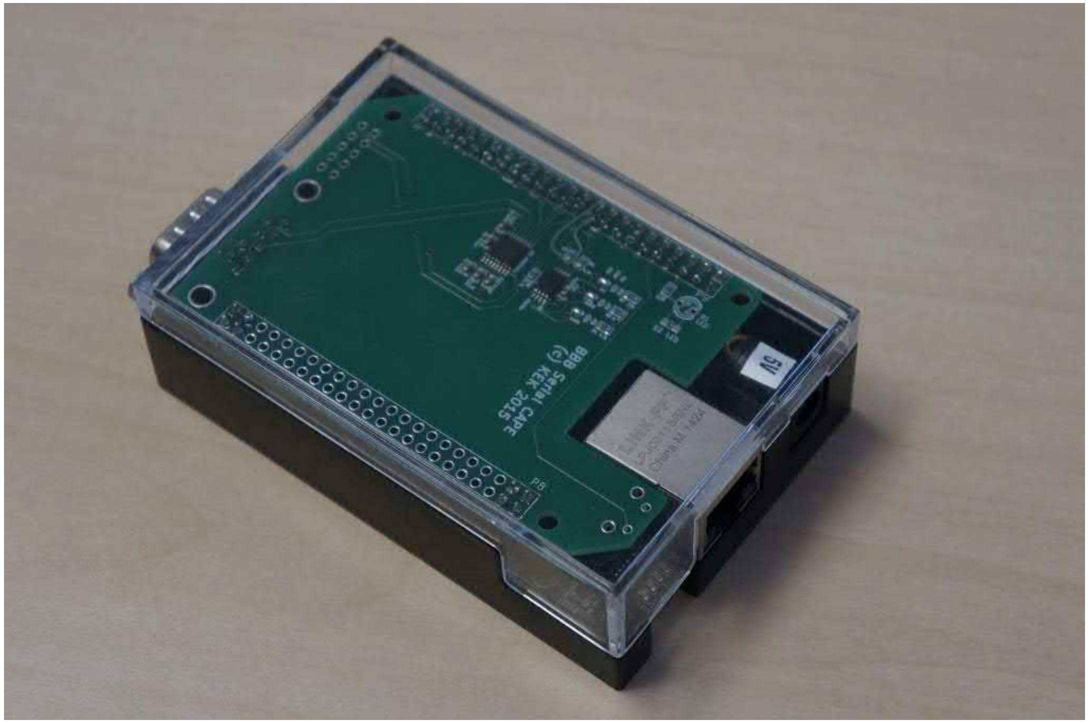
- Advantage of Raspberry Pi
  - Most popular
  - Many information is available on the web. Many books on the market.
  - Variation : RasPi Zero, Zero-W, Compute Module
  - Good combination with Arduino
- Disadvantage of Raspberry Pi
  - No analog input (need Arduino)
  - Connector location is not good for embedding
  - High power consumption (Pi3)



RS-232 Device Server (KEK prototype)

Purpose : Replace Ethernet-RS232 converter with EPICS IOC box.

- BBB + Cape
- Board designed by ourselves (Michikawa-san)
- Cheap! Working stably for long time.

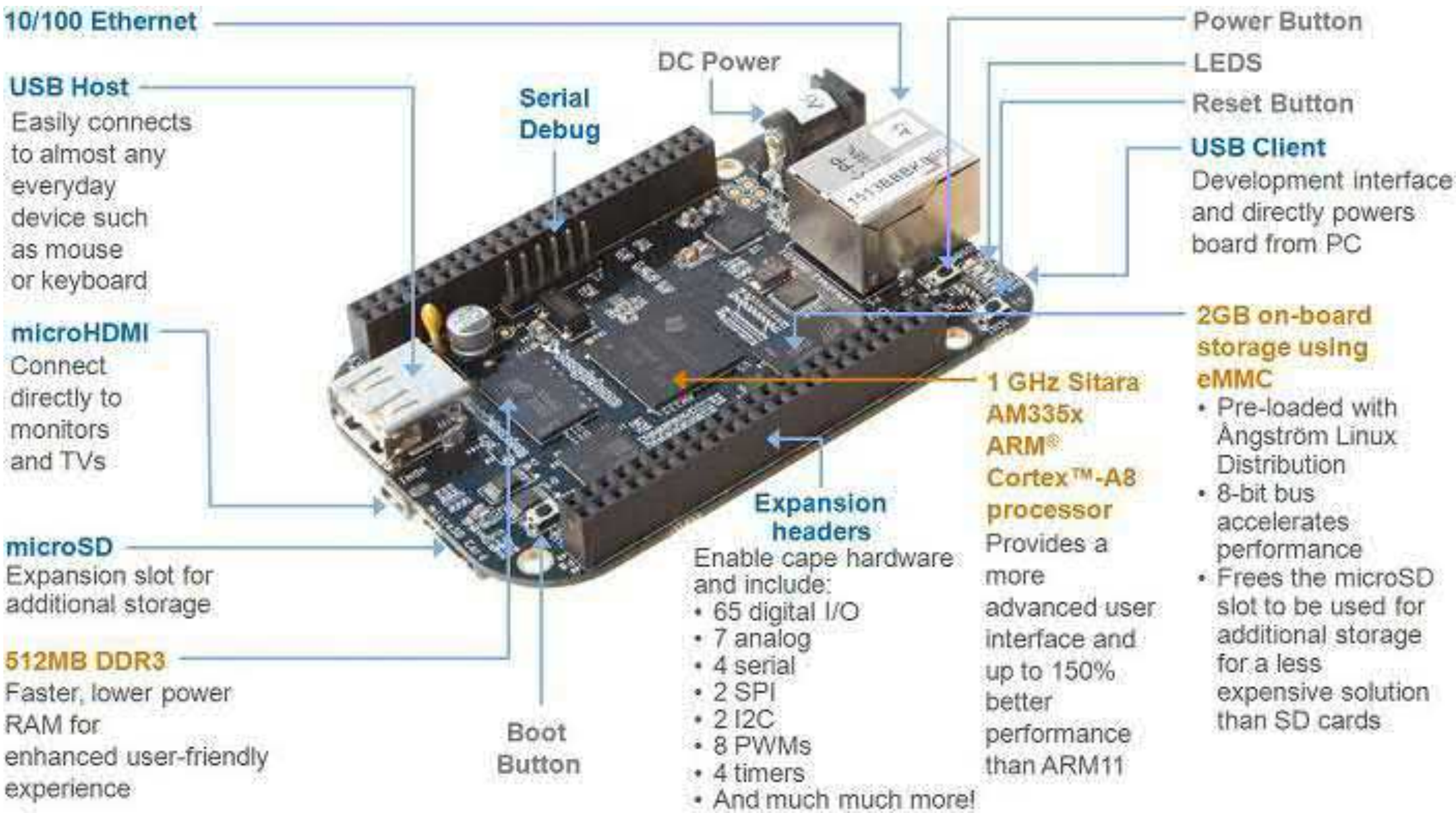


My personal opinion : BeagleBone Green is best for EPICS IOC because we don’t use it as GUI terminal (in many cases). A laptop PC is much powerful than BBB or RasPi.

Step 1: Hardware

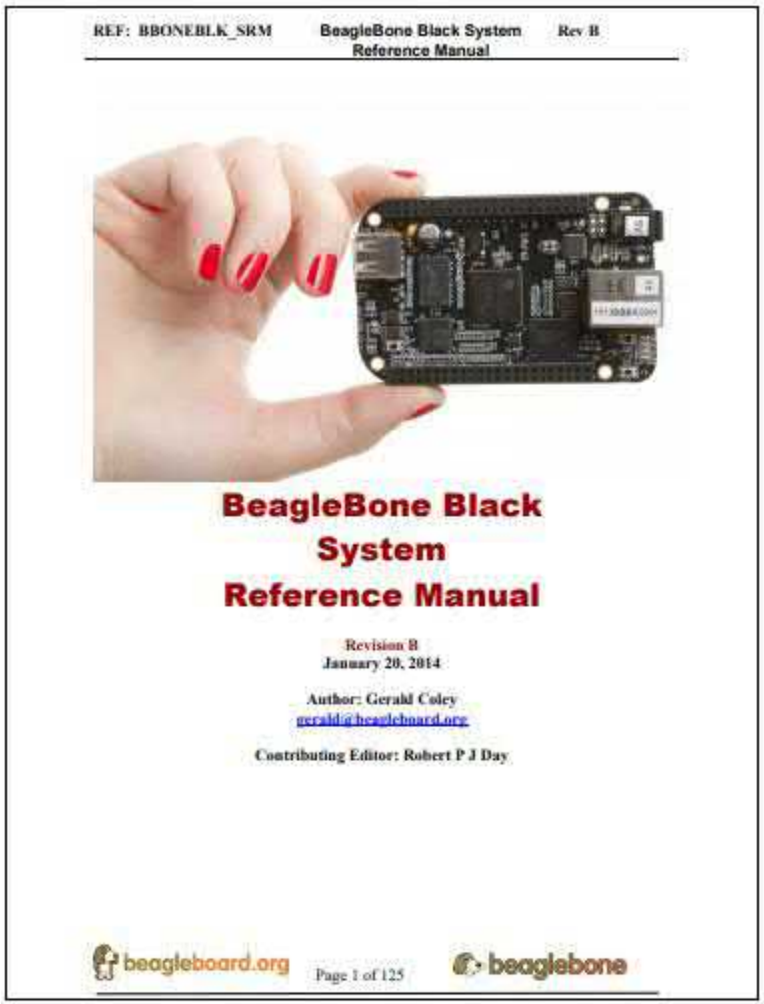
- Check Hardware
  - BeagleBone Black and/or BeagleBone Green

http://www.tij.co.jp/tool/jp/beaglebk



System Reference Manual (SRM)

https://github.com/CircuitCo/BeagleBone-Black/blob/rev_b/BBB_SRM.pdf


REF: BBONEBLK_SRM — BeagleBone Black System Reference Manual — Rev B

BeagleBone Black System Reference Manual

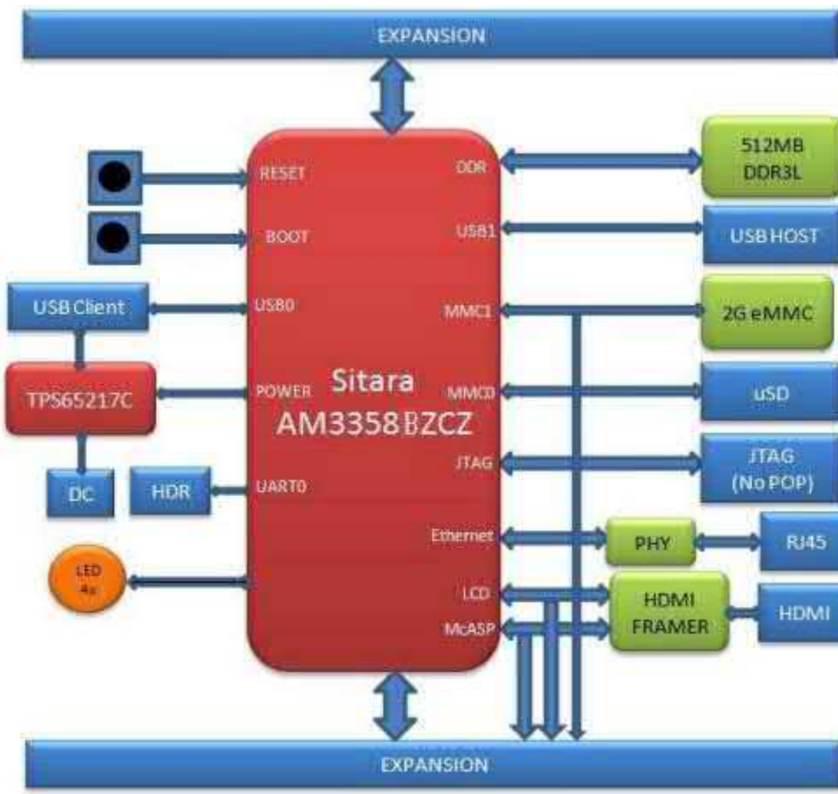
Revision B
January 20, 2014

Author: Gerald Coley
gerald@beagleboard.org

Contributing Editor: Robert P J Day

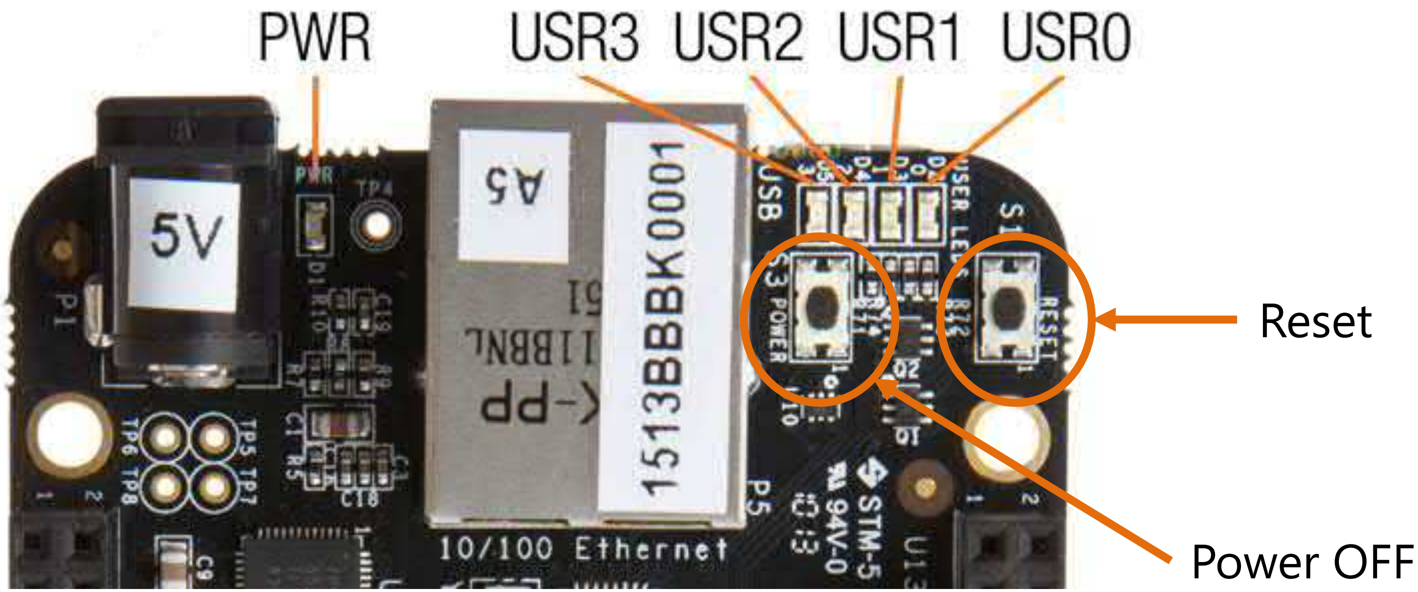
beagleboard.org — Page 1 of 125 — beaglebone

Key Components



on-board LED, Button

http://beagleboard.org/getting-started



USR0 is typically configured at boot to blink in a heartbeat pattern
USR1 is typically configured at boot to light during SD (microSD) card accesses
USR2 is typically configured at boot to light during CPU activity
USR3 is typically configured at boot to light during eMMC accesses

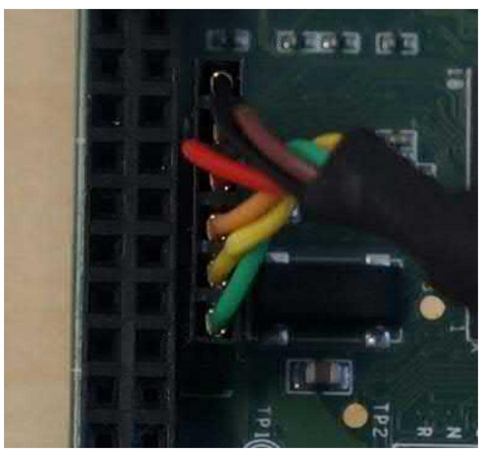
You can configure the LED behavior (check it later).

Power ON, Check Boot Sequence

Power feed : DC 5V Jack or USB

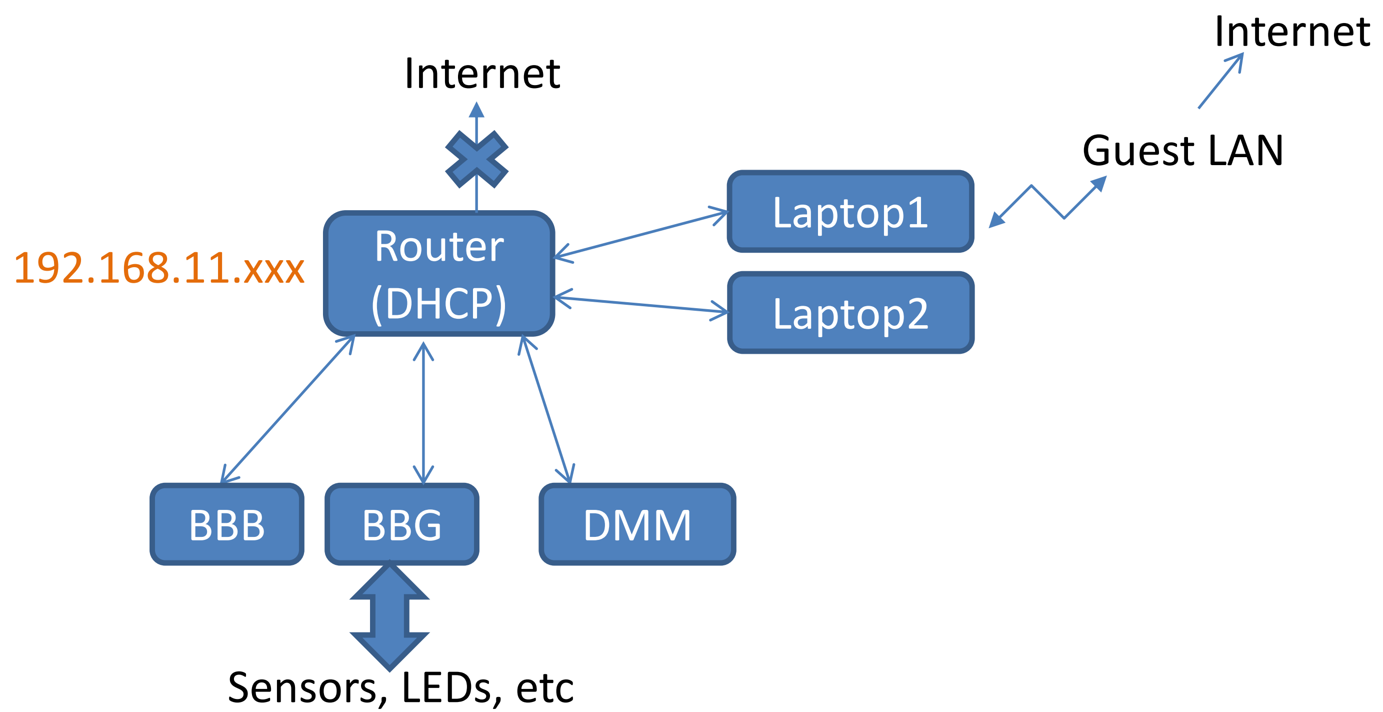
Connect Serial Console (demo)

- 3.3 V USB to Serial cable (FTDI chip); PIN1 ➔ GND(Black)
- Terminal Software (Tera Term, Putty, etc), 11520bps, 8bit, Parity None
- Download windows device driver from FTDI web site (skip today)



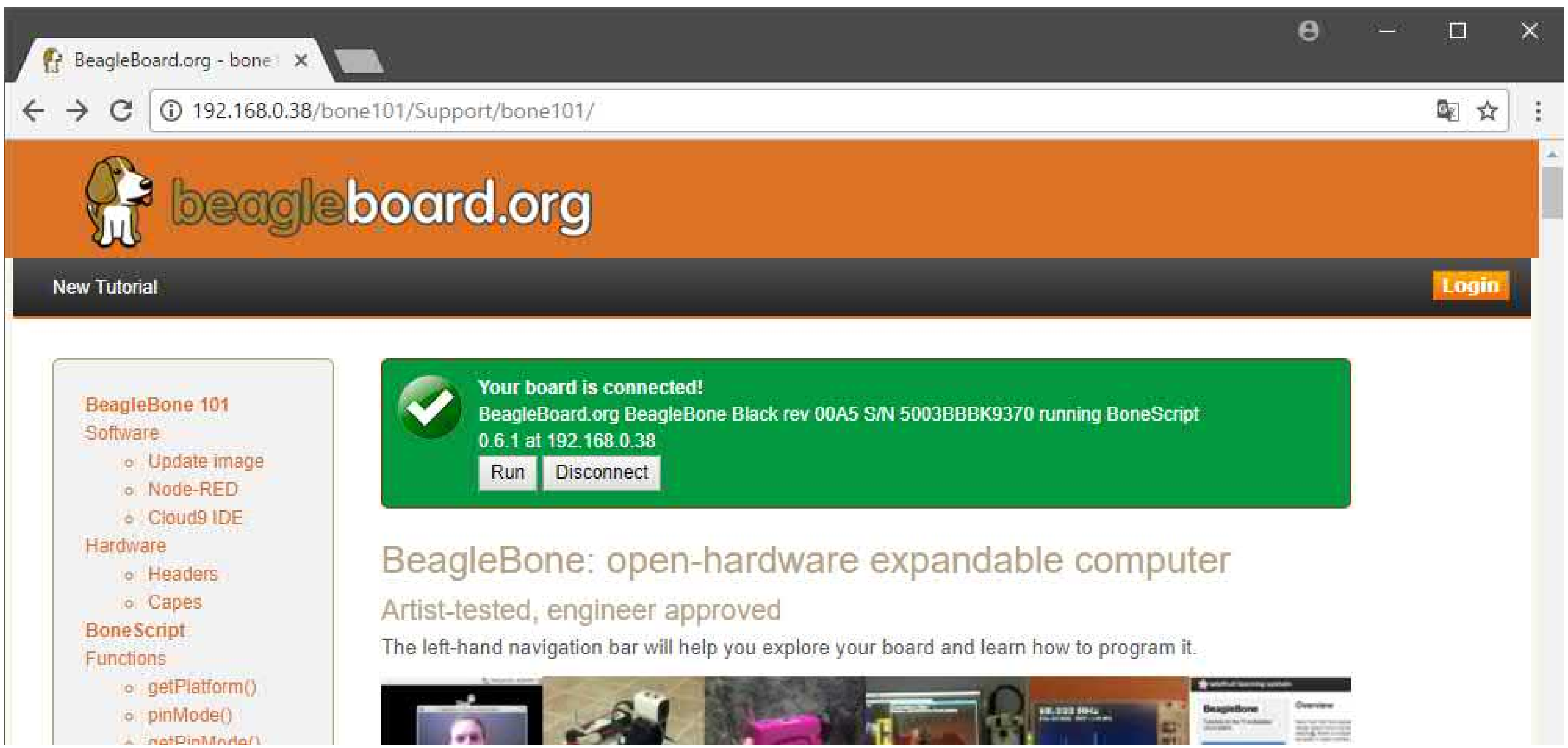
Step 2 : Network

- Network configuration
- Check IP address of your PC and BBB/BBG
- ping to BBB
- ssh to BBB
- http connection (web browser) to BBB (next page)
  - check routing table



connect via http

- check for connection
- It is possible to use “Bonescript” to control BBB
  - we don’t use them
- This page is useful to check “Pinout” configuration



Further Learning

network command : basics

- ping
- netstat
- traceroute
- arp

Learn IP address, subnet, MAC address

Ping to subnet hosts, and check arp table

- write a python script to do that

Step 3 : using GPIO

GPIO : General-Purpose Input/Output port

- Check Pinout of BBB (P8, P9 header)
- Learn usage of solderless breadboard

Procedure of the lesson

A) Digital Output : Turn LED on/off with GPIO
B) Digital Input : Button switch to process EPICS record (I/O Intr)

Pinout Header

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	GPIO_112
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND

P8

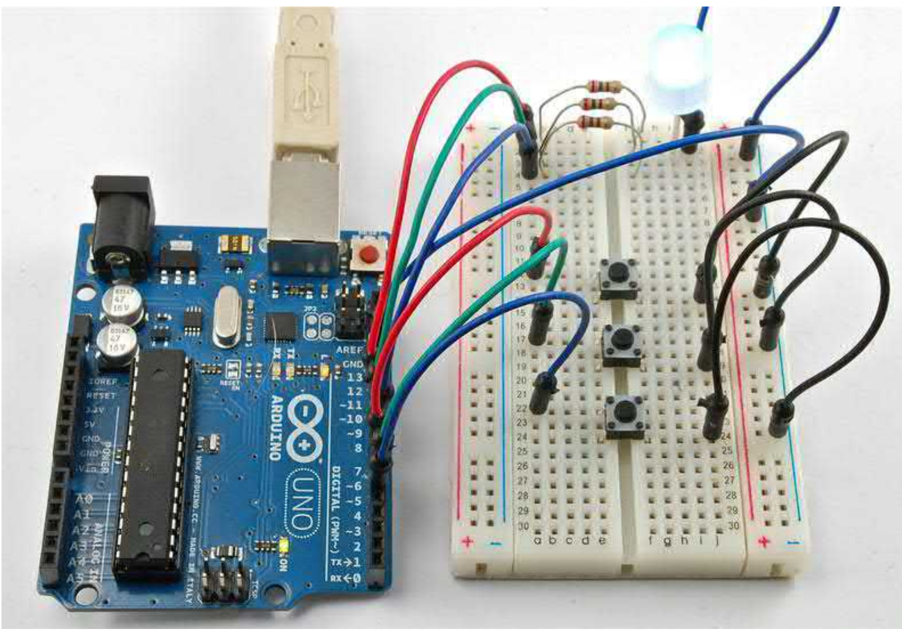
DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

LEGEND
Power/Ground/Reset
Available Digital
Available PWM
Shared I2C Bus
Reconfigurable Digital
Analog Inputs (1.8V)

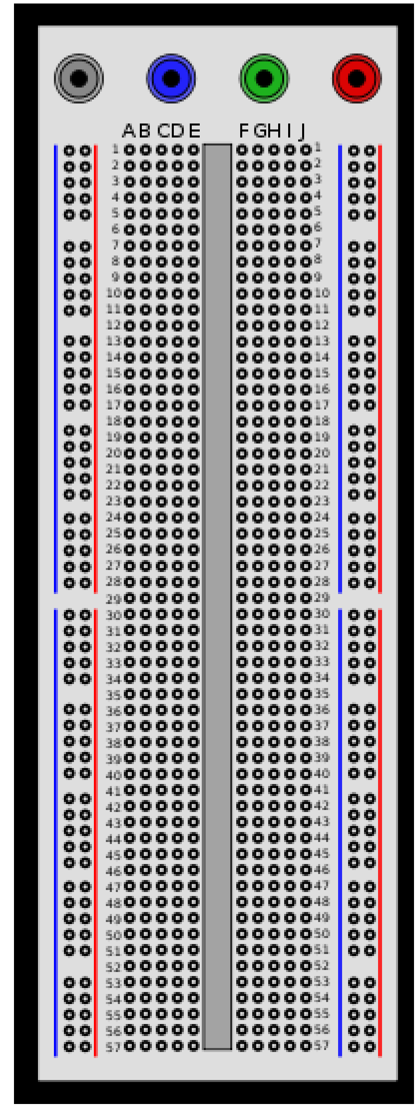
Solderless Breadboard

Very convenient tool for prototyping.

- Use Jumper-wire to connect
- horizontal line(A-E, F-J) is connected internally
- place ICs in the center (Between E-F in right fig.)



https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard
https://learn.adafruit.com/breadboards-for-beginners



wikipedia

Lesson A) Digital OUTPUT

Lesson Procedure

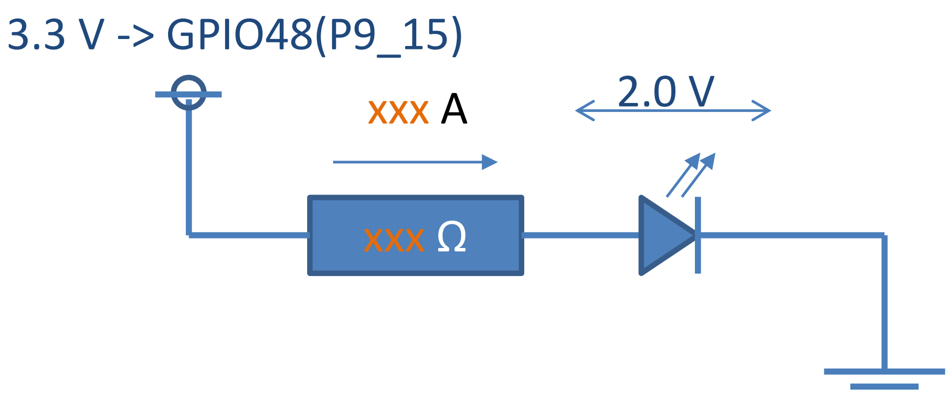
1. Connect a LED and a resister as shown in the next page
   - Don’t forget toad current limiting resister to protect BBB !!
2. Test with SYS_3.3V (P9_ 3 or 4) to check polarity and health of LED.
3. connect GPIO pin
4. Turn ON/OFF with python script
5. Turn On/OFF with EPICS

copy sample program from /opt/epics/sample to your home directory, and edit

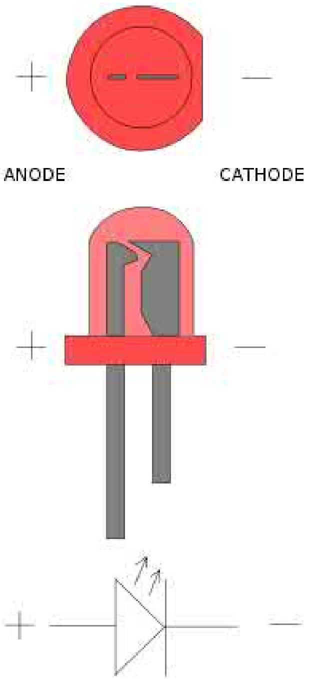
Current limiting resister

To turn on a LED, you need to calculate "current" and "voltage"

- Typical Red LED need at least 2 V (check datasheet for detail)
- Recent high brightness LED need at least 2 – 3 mA (check max current)



LED (Wikipedia)



Note on BBB Hardware Specification

- GPIO pin is 3.3 V
- Max 4 – 5 mA on each port
- Total (about) 10 mA for all GPIO pins
- SYS_5V and SYS_3.3V can source more current (~250 mA)

Lesson A) Control Digital Input from a program

1. Use python + Adafruuit_BBIO module
   - Good tool for check

2. Execute an “Example” Soft IOC
   - detail in next page

3. Use EPICS (devgpio)
   - create database
   - execute st.cmd as root privilege (sudo)
   - use dbpf to control
   - use caput to control
   - create CSS or medm GUI to control ON/OFF (Optional)

EPICS Example Application

EPICS software IOC

- Build an "example application" on BBB
  - makeBaseApp.pl –l ← list template
  - makeBaseApp.pl –t example example
  - makeBaseApp.pl –i –t example example
  - make ← check errors
  - edit exampleApp/Db/user.substitutions ← replace "debian" to your name
  - make
  - cd iocBoot/iocexample/
  - edit st.cmd ← replace "debian" to your name
  - chmod +x st.cmd
  - sudo ./st.cmd
- execute st.cmd, and use camonitor to check value
- Create a MEDM or CSS panel, and monitor aiExample1
- monitor PVs on the other BBB or BBG

Need to install VcXsrc or Xming to use MEDM or Gnuplot

EPICS Example Application

Optional topics

- Use “excas” to execute simple soft ioc (jane, fred, freddy, etc)
- camonitor the value, and redirect to some file
- Use gnuplot to visualize the data

Need to install VcXsrc or Xming to use MEDM or Gnuplot

timestamp of BBB

Ubuntu on BBB automatically adjust timestamp via NTP

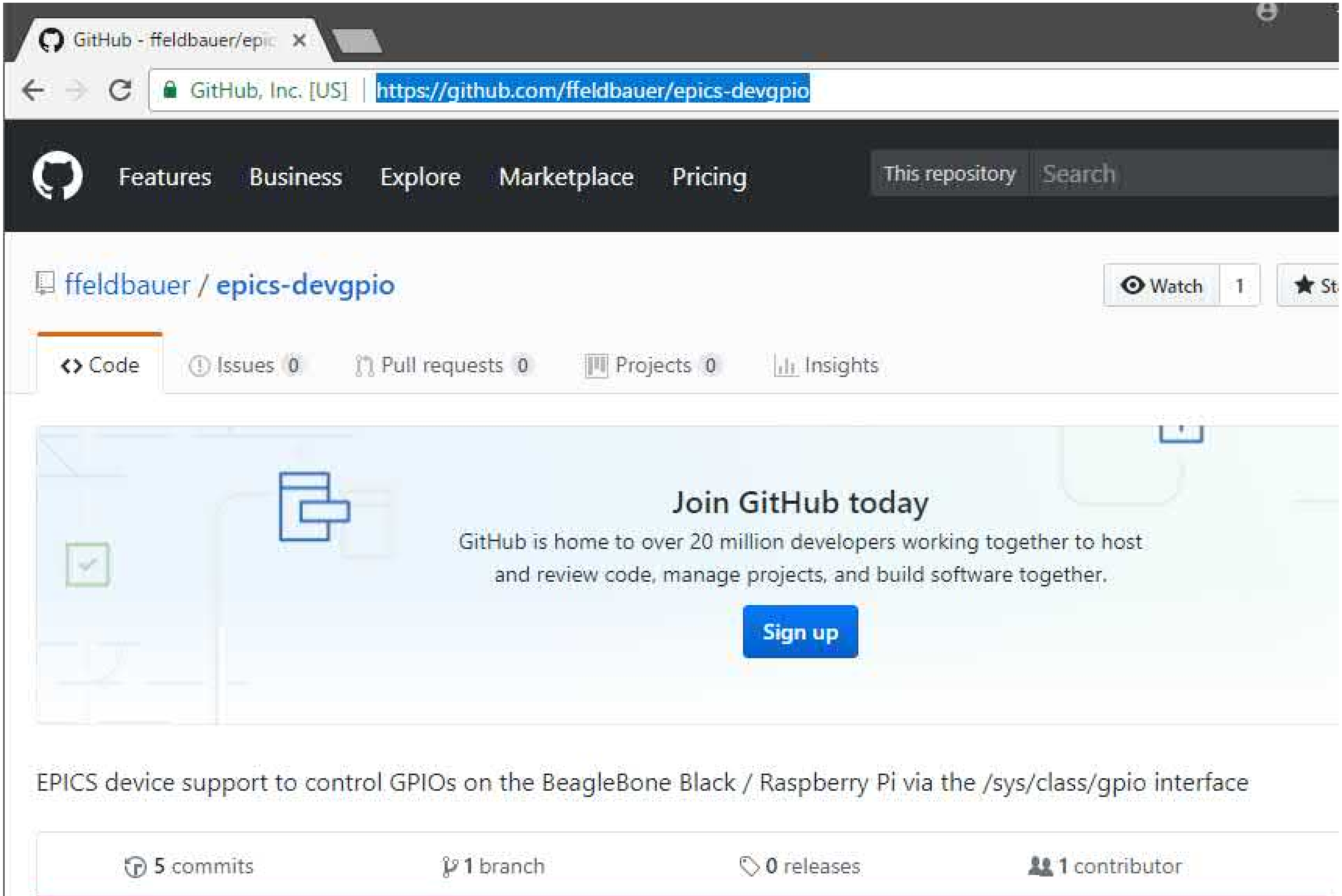
- systemctl -l status systemd-timesyncd

BBB (and RasPi) do not have RTC (real-time clock) and separate battery

- Today's network is isolated from internet
- Just adjust the time manually. i.e. <u>sudo date –s '2018/01/30 15:00:00'</u>
- otherwise time difference between your laptop and BBB will cause problem

EPICS devgpio

https://github.com/ffeldbauer/epics-devgpio



Lesson B) Digital Input with EPICS

Digital Input

- edit “gpio1.db” -> add your name on the head
- use EPICS devgpio
- add a switch (SPST) on breadboard. Press button for high level
- No resister is required on the input pin ※
- use I/O interrupt to process EPICS record

※ Note

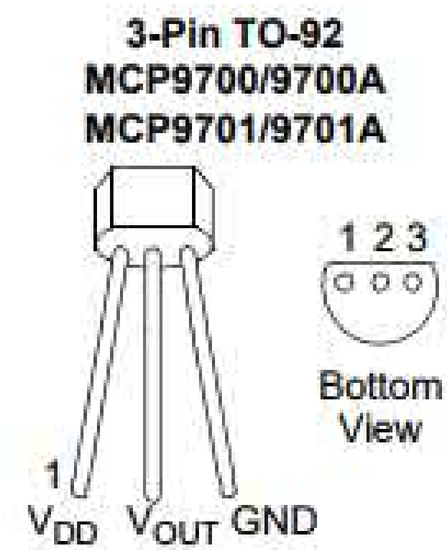
- Internal pull-down resister exists. (Configurable!!)
- Open Pin → low level
- Consider the difference of “Source/Sink”
- How to measure the resistance of internal resister? Pull-up or Pull-down?

Step 4 : Analog Input

Temperature Sensor

- MCP9701
- 3 wire, analog output sensor
- Cheap : ¥40 - ¥50

- 400 mV at 0 °C
- Tcoef = 19.5 mV/°C
- Vout = Tcoef * Temp + V_0deg



Further (Advanced) Topics

Advanced topics : Hardware

- Serial interface (UART, i2c, SPI, 1-wire, HID device, etc)
  - Many sensors can be attached
- Analog Input
  - Attach voltage limiting circuit (Op-Amp) or Voltage divider to extend input voltage range from 0-1.8V to 0-5 V, +/-10V, for example.
  - Test other equipment : Magnetic sensor (Hall effect sensor), Joystick, etc
- PWM (can be used instead of analog output)
  - Drive servo motors or LEDs
- Drive Stepping Motor
- Use Arduino for analog and sensor input
- RPU (Real-time Processing Unit) of BBB
- Other ARM-Linux based board : Tinker board, RPi3, Armadillo, etc
- Boot from eMMC

Advanced topics : Software related

- Realtime OS, Create custom micro-SD boot image, etc

Further (Advanced) Topics : EPICS

Improvement of the hands-on/training procedure

- document, hardware, procedure, etc

EPICS related topics

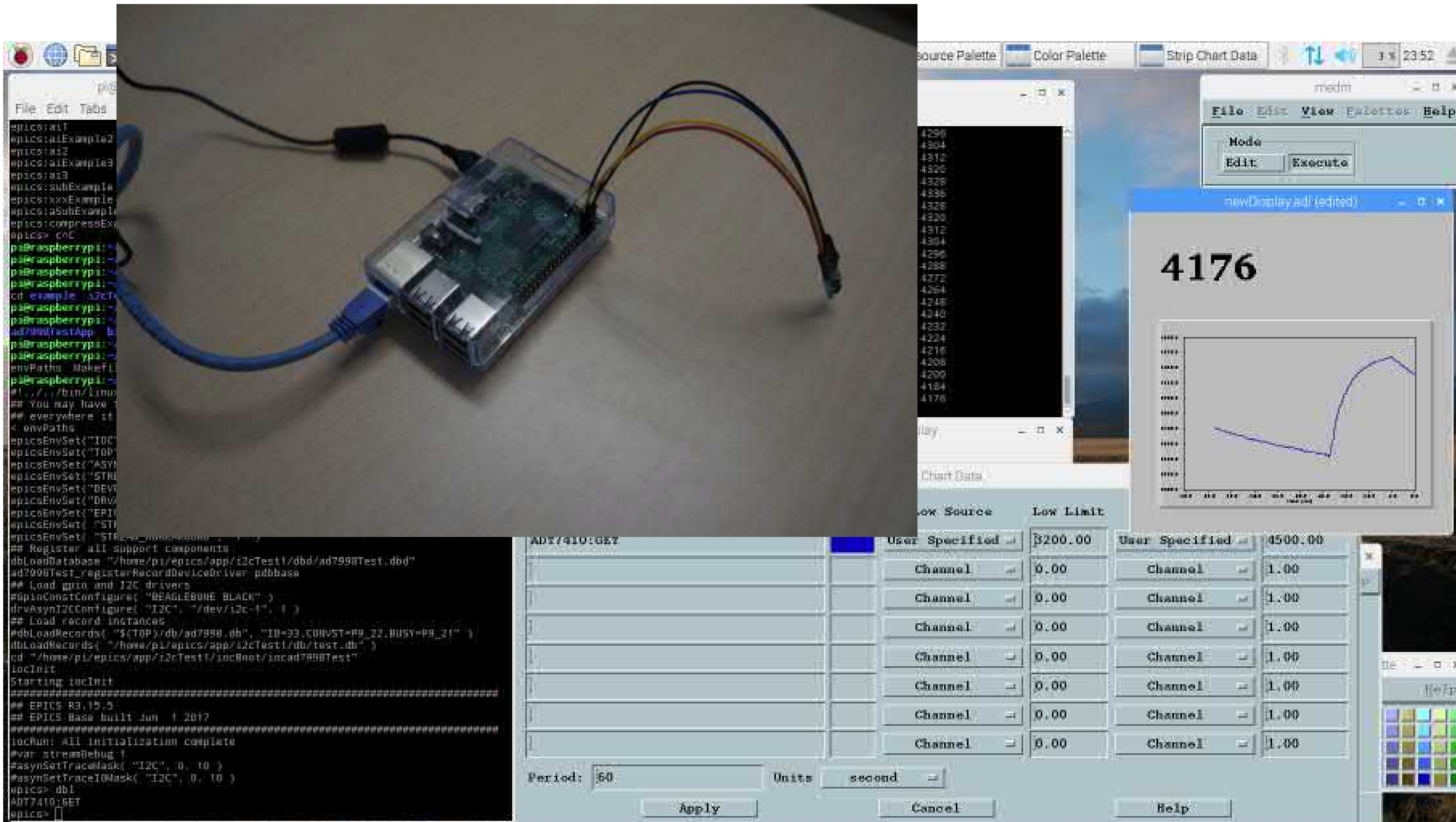
- Asyn + Stream device (for example, DMM via Ethernet)
- Build EPICS Base/Extensions/Modules, Device support, etc
- Python/CA interface (pyepics or PythonCA)
- Development of device support
- I/O interrupt : How fast?
- Cross-compile environment

- Archiver on BBB
  - CSS Archiver, Archiver Appliance, Channel Archiver, Portable Archiver

Example : i2c temperature sensor

http://cerldev.kek.jp/trac/EpicsUsersJP/wiki/epics/raspberrypi/setup_epics_i2c

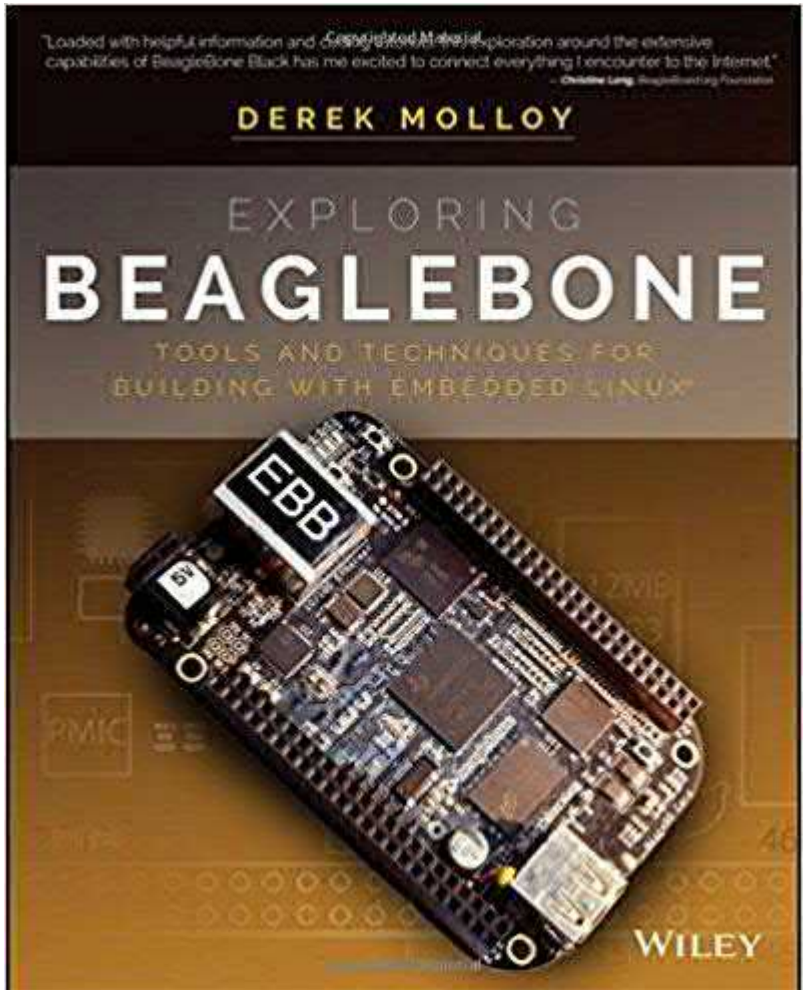
- ADT7410 Temperature Sensor (Analog Devices)

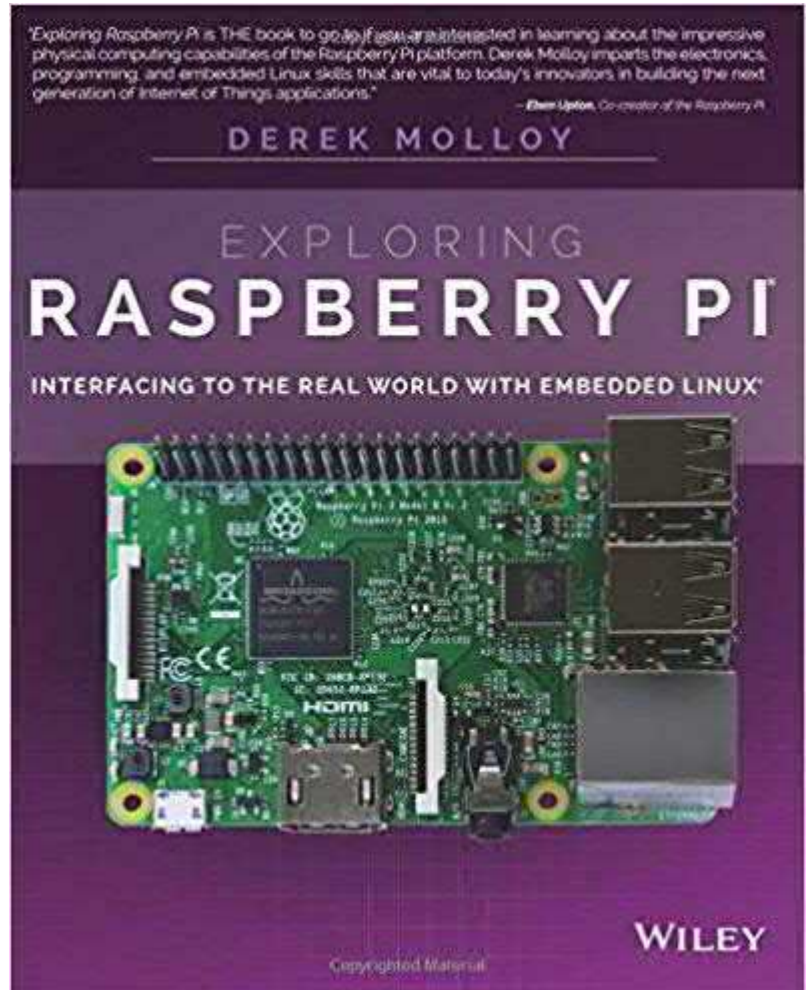


References

Reference : Books

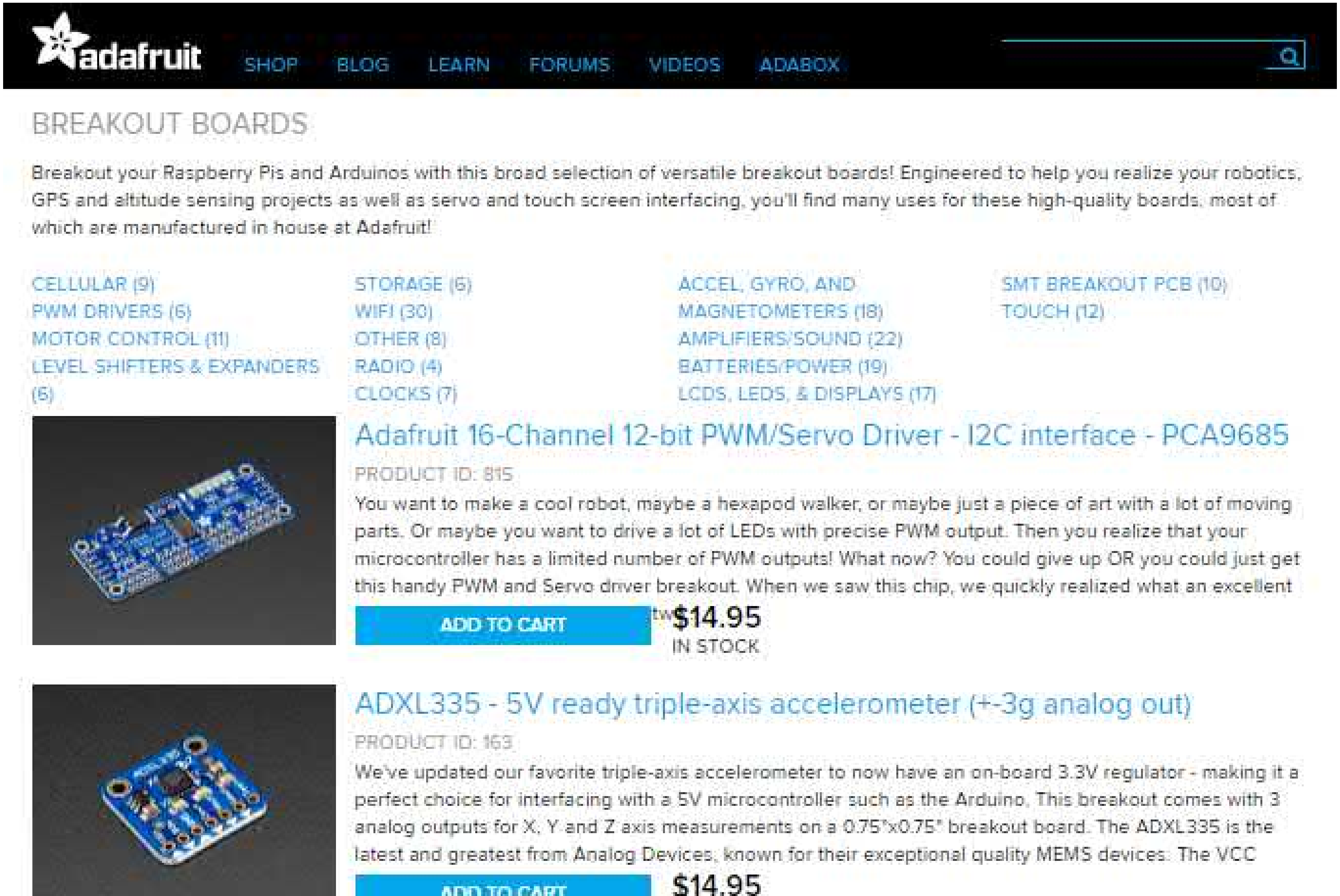
- http://exploringbeaglebone.com/





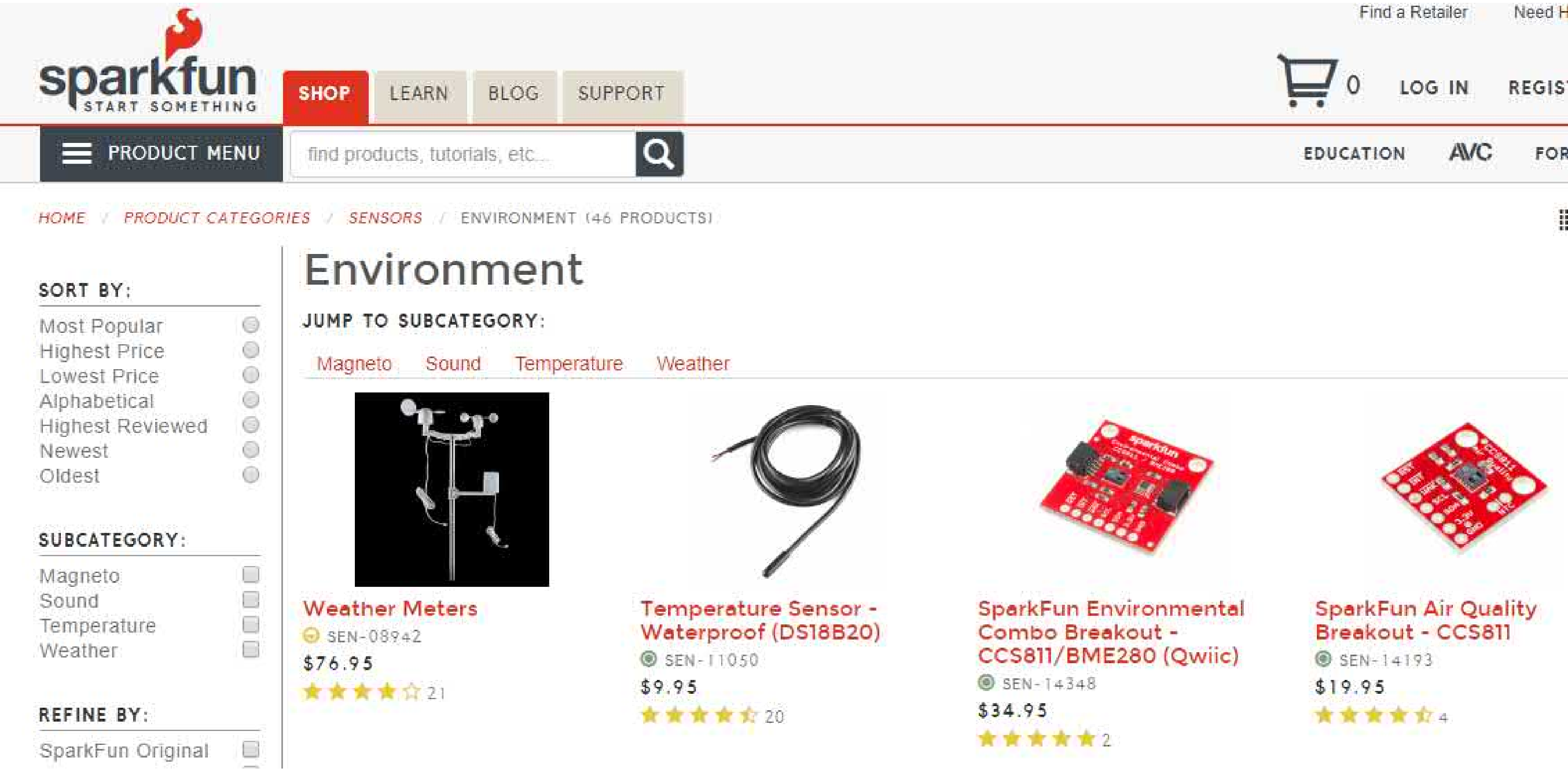
Adafruit

- https://www.adafruit.com/



Sparkfun

- https://www.sparkfun.com



秋月電子通商

- http://akizukidenshi.com

千石電商

- https://www.sengoku.co.jp/

shop, etc

Online Shop

– RS Components	https://jp.rs-online.com/web/
– Digi-key	https://www.digikey.jp/
– Mouser Electronics	https://www.mouser.jp/

Parts/Kit shop

– Adafruit	https://www.adafruit.com/
– Sparkfun	https://www.sparkfun.com
– Akizuki(秋月)	http://akizukidenshi.com
– Sengoku(千石)	https://www.sengoku.co.jp/
– Switch Science	https://www.switch-science.com/
– marutsu parts shop	https://www.marutsu.co.jp/
– Strawberry Linux	https://strawberry-linux.com/

Link

Web site

- BeagleBone Black https://beagleboard.org/black
- Raspberry Pi https://www.raspberrypi.org/