

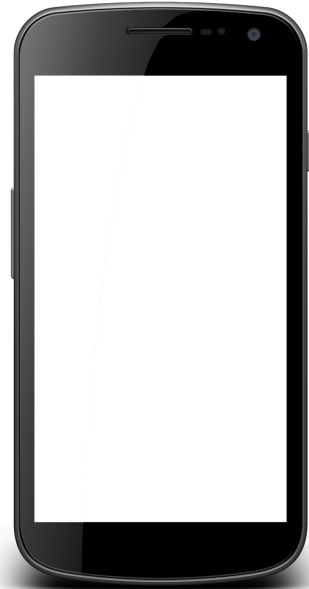
Experimentações da Internet das Coisas

#04 Serial e Bluetooth

Ivanovitch Silva
Agosto, 2017



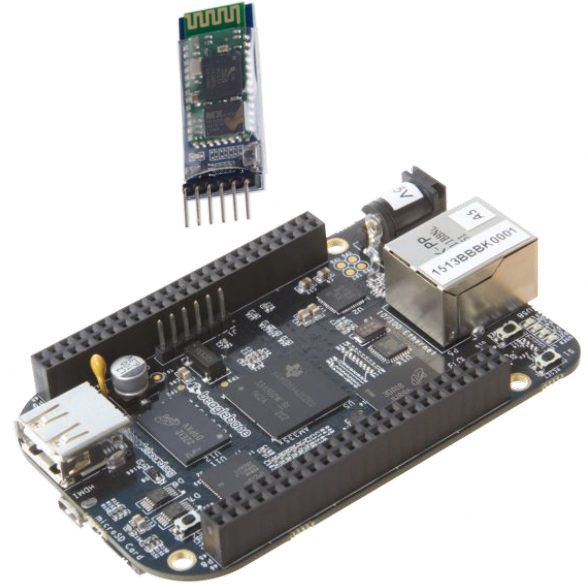
Qual o objetivo agora?



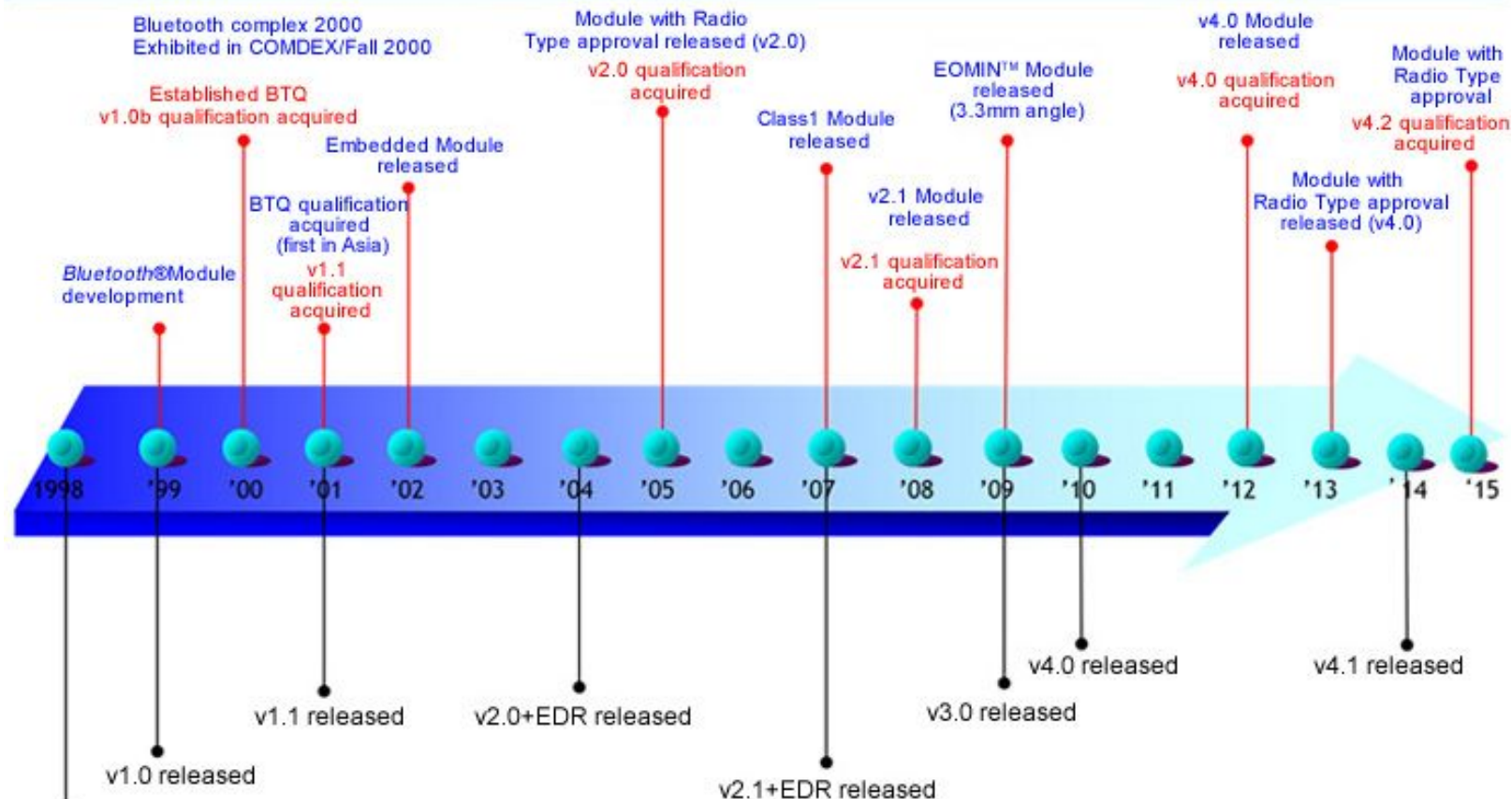
App genéricas
"Bluetooth Terminal"



HC Family



Sensores &
Atuadores

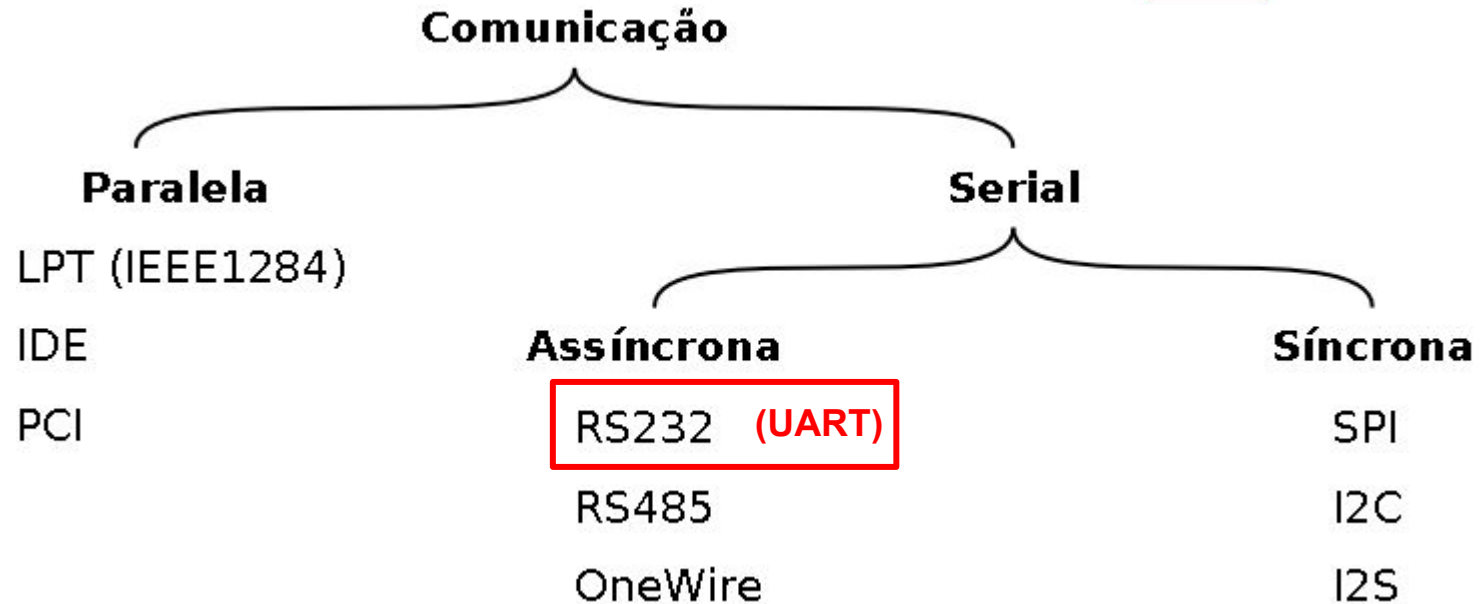
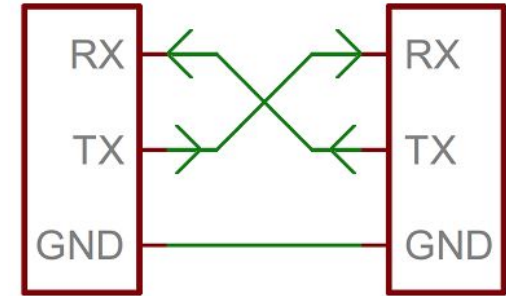


Established Bluetooth® SIG

Bluetooth® History

Comunicação Serial

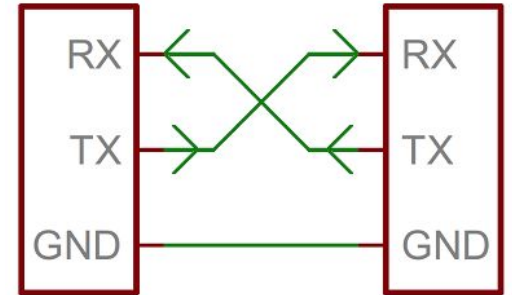
Tecnologias de comunicação

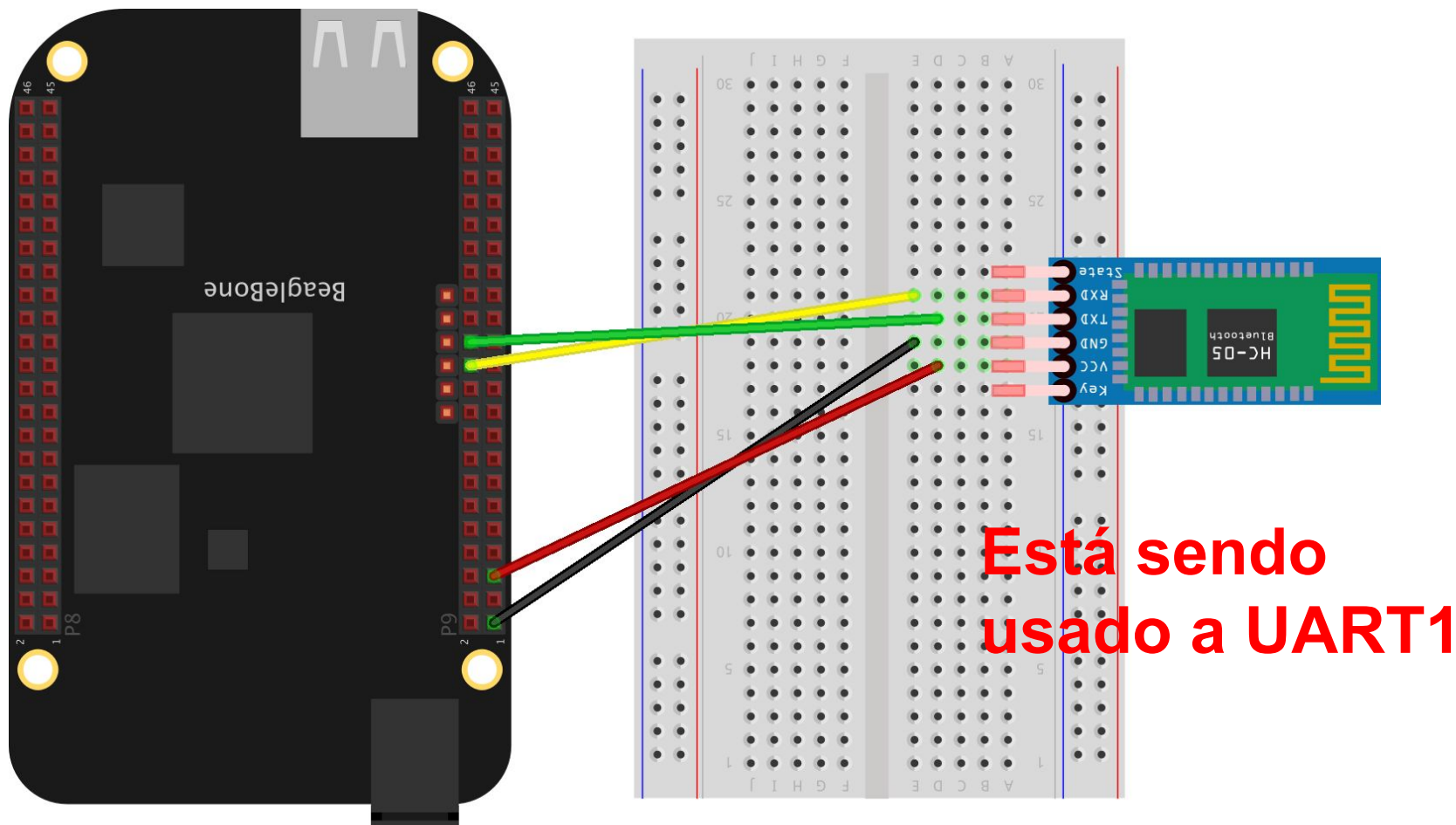


4 UARTs and 1 TX only

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BUT	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
UART4_RXD	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
UART4_TXD	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
UART1_RTSN	19	20	UART1_CTSN	GPIO_22	19	20	GPIO_63
UART2_TXD	21	22	UART2_RXD	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	UART1_TXD	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	UART1_RXD	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	GPIO_113	GPIO_86	27	28	GPIO_88
GPIO_111	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
GPIO_110	31	32	VDD_ADC	UART5_CTSN+	31	32	UART5_RTSN
AIN4	33	34	GND_A_ADC	UART4_RTSN	33	34	UART3_RTSN
AIN6	35	36	AIN5	UART4_CTSN	35	36	UART3_CTSN
AIN2	37	38	AIN3	UARR5_TXD+	37	38	UART5_RXD+
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	UART3_TXD	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	GPIO_70	45	46	GPIO_71

HC - 05





```
$sudo config-pin -q P9.24  
$sudo config-pin -q P9.26
```

```
$sudo config-pin P9.24 uart  
$sudo config-pin P9.26 uart
```

fritzing

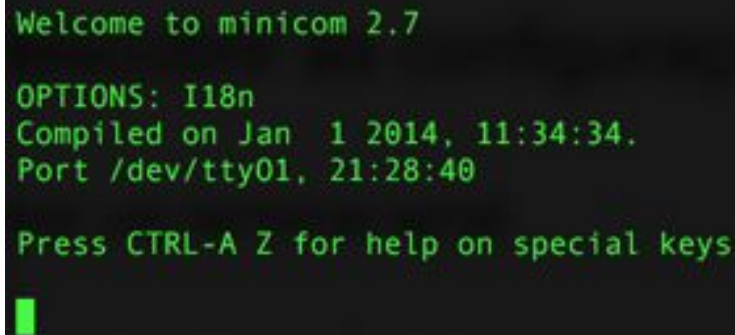
Gerenciando as configurações do HC-05

1. Identificar o tipo HC-05 (com ou sem botão)
 - a. Com botão: ligar o 5V ao mesmo tempo que o botão estiver pressionado. Após isso soltar o botão.
 - b. Sem botão: alimentar o pino EN com 3.3V
 - c. No modo configuração o Bluetooth não funciona, apenas comandos AT são suportados.
 - d. Escrever um programa ou usar algum terminal (e.g minicom) para enviar comandos AT
2. Dicas: <https://goo.gl/z1kRcM>

Como descobrir as configurações do HC-05?

- Instalar um terminal serial

`sudo apt-get install minicom`



```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan 1 2014, 11:34:34.
Port /dev/ttyO1, 21:28:40

Press CTRL-A Z for help on special keys

█
```

Dica (visualizar o comando): Ctrl+A
Z
E

```
$minicom -b 38400 -o -D /dev/ttyO1
```

Comandos AT (HC-05)

Fonte: http://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth_hc05.pdf

- AT
- AT+RESET
- AT+VERSION?
- AT+ORGL
- AT+ADDR?
- AT+PSWD?
- AT+NAME=SEUNOME

Construção do experimento [Passo 1]

1. Instalar APP Android Genérico "Terminal Bluetooth"
 - a. <https://play.google.com/store/apps/details?id=project.bluetoothterminal&hl=en>
 - b. <https://play.google.com/store/apps/details?id=ptah.apps.bluetoothterminal&hl=en>
2. Fazer a sincronização ("pareamento") entre o celular e o HC-05
 - a. [No celular] Settings > Bluetooth >
3. Abrir o APP recém instalado e conectar ao HC_05

Programando...



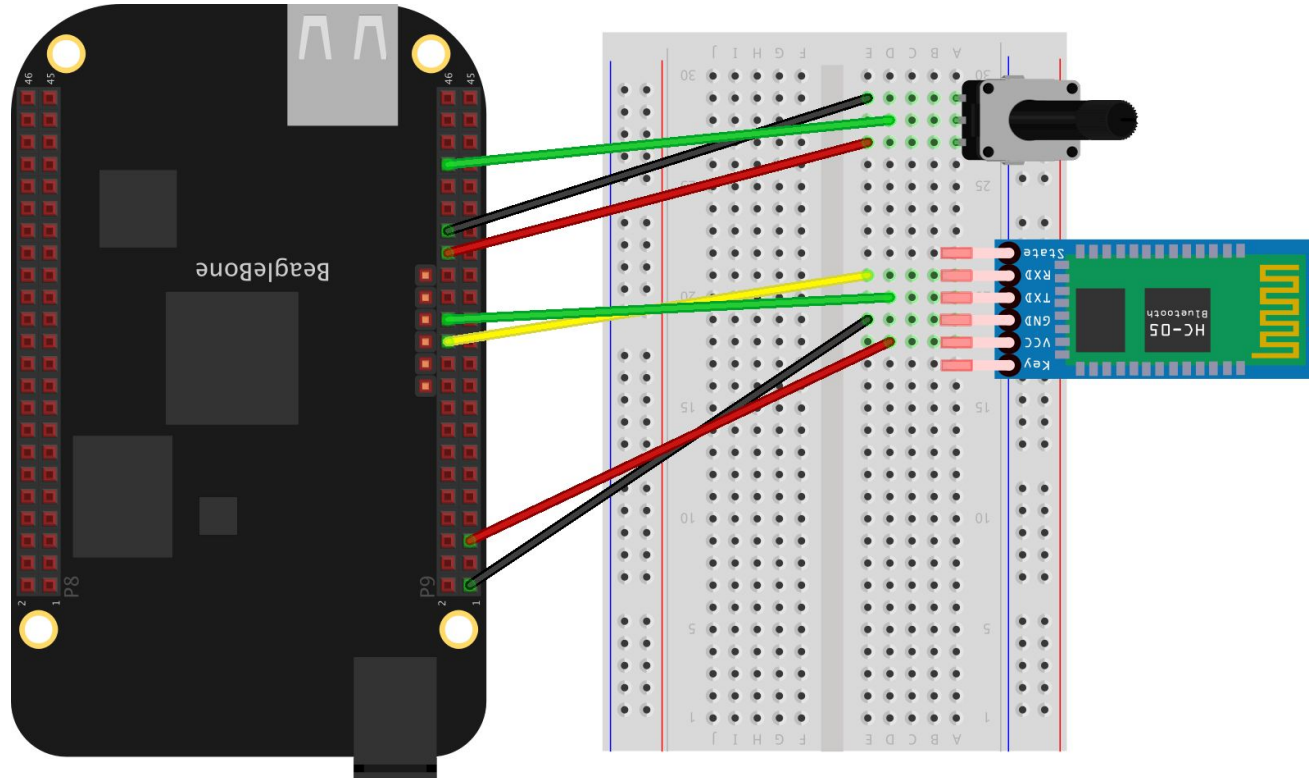
```
import Adafruit_BBIO.UART as UART
import serial

UART.setup("UART1")

ser = serial.Serial(port = "/dev/tty01", baudrate=9600)
if ser.isOpen():
    print("Serial is open!")
    ser.write("Hello World!")
ser.close()
```

\$sudo python helloworldserial.py

Enviando dados do potenciometro para o celular



Enviando dados do potenciometro para o celular

```
import Adafruit_BBIO.ADC as ADC
import Adafruit_BBIO.UART as UART
import serial

UART.setup("UART1")
ADC.setup()

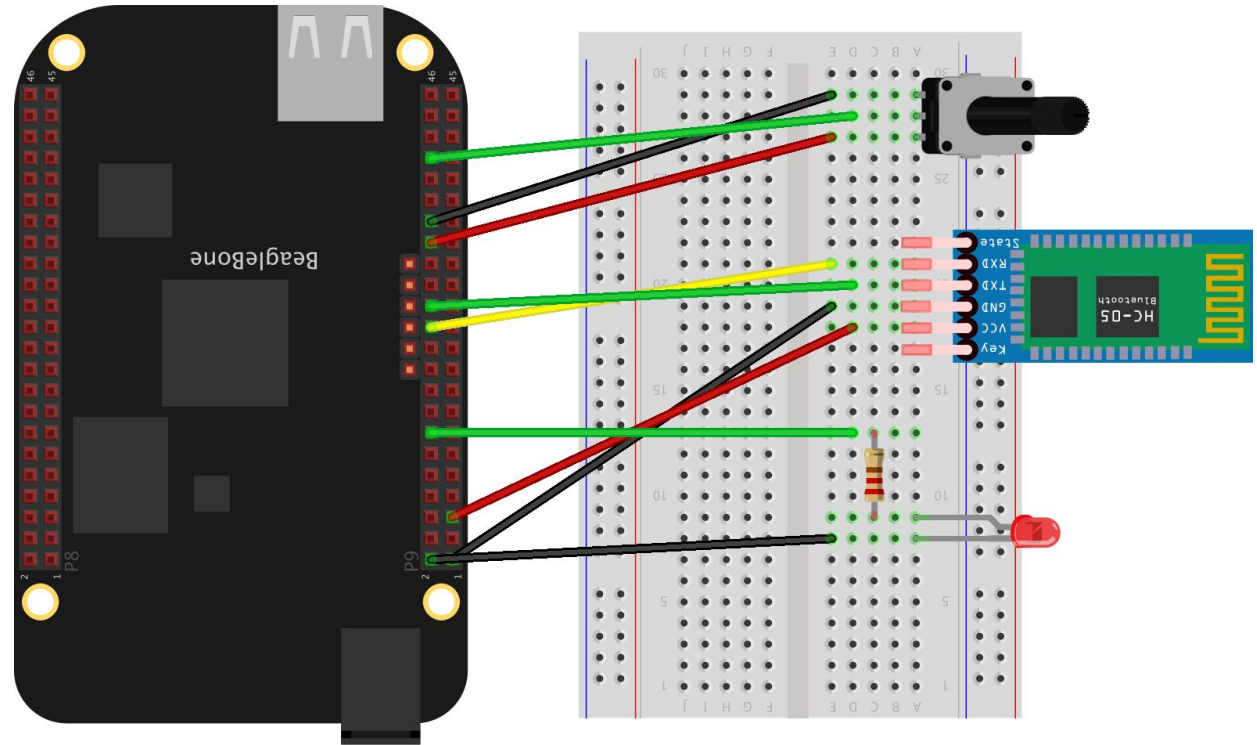
ser = serial.Serial(port = "/dev/tty01", baudrate=9600)
value = ADC.read_raw("P9_40")

if ser.isOpen():
    print(value)
    ser.write(str(value) + "\n")
ser.close()
```

\$sudo python potenciometroSerial.py

Experimento

- Ligar o Led
 - Digitar "on"
- Apagar o Led
 - Digitar "off"



Fim da aula #4