

```
In [1]: #from pynq.overlays.base import BaseOverlay
#from pynq.lib.arduino.arduino_analog import Arduino_Analog
#import matplotlib.pyplot as plt
#from time import sleep

#ol = BaseOverlay('base.bit')
#xadc = Arduino_Analog(ol.ARDUINO, [5])
# Select the sampling interval in the unit of ms.
#xadc.set_log_interval_ms(1)
#xadc.start_log()
# Wait for stacking data.
#sleep(1)
#v = xadc.get_log()
# print(v[0])
```

```
In [2]: import time
from pynq.overlays.base import BaseOverlay
base = BaseOverlay("base.bit")
from pynq.lib.arduino import Arduino_Analog
from pynq.lib.arduino import ARDUINO_GROVE_A1

analog0 = Arduino_Analog(base.ARDUINO, ARDUINO_GROVE_A1)
a = []
t_start = time.time()
t_end = time.time() + 1
while time.time() < t_end:
    a.append(analog0.read()[0])
```

```
In [4]: %%matplotlib inline
import matplotlib.pyplot as plt
from matplotlib.legend_handler import HandlerLine2D

plt.rcParams['figure.figsize'] = [10, 12]
plt.rcParams['figure.dpi'] = 100 # 200 e.g. is really fine, but slower

#line1, = plt.plot(range(len(a)), a,
#                  'ro', Label="Voltage on ADC")
#plt.title('Analog to Digital Converter')
#plt.axis([0, len(a), 0.0, 3.3])
#plt.xlabel('Sample number')
#plt.ylabel('Voltage')
#plt.legend(loc=4, bbox_to_anchor=(1, -0.3),
#          ncol=2, borderaxespad=0.,
#          handler_map={line1: HandlerLine2D(numpoints=1),
#                      })
plt.show()
plt.savefig('adc.pdf', bbox_inches='tight')
```