gpio via C

- Accessing `/sys` through C
- Using `usleep()`

- Review:
  - togglegpio.c
  - gpio-utils.c
  - gpio-utils.h

gpio via C–Interrupts

- Accessing /sys through C
- Using poll()

- Review
    - gpio-init-test.c

libsoc

- C library for interfacing with common System on Chip (SoC) peripherals through generic kernel interfaces (https://github.com/jackmitch/libsoc)
- Aimed at new Linux users
  - intends to be a stepping stone to enable a user to get started quickly.
  - optimized for reliability rather than speed. While the library should be fast, no guarantees are made on it's determinism and it should not be used in time critical routines.

Installing

- Already installed on Bone

libsoc Example

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

#include <libsoc_gpio.h>
#include <libsoc_debug.h>

// Blinks an LED attached to P9_14
#define GPIO_OUTPUT  50
```

libsoc Example

```
int main(void) {
  gpio *gpio_output;    // Create gpio pointer
  libsoc_set_debug(1);  // Enable debug output
  // Request gpio
  gpio_output = libsoc_gpio_request(GPIO_OUTPUT,
                                        LS_SHARED);
  // Set direction to OUTPUT
  libsoc_gpio_set_direction(gpio_output, OUTPUT);

  libsoc_set_debug(0);   // Turn off debug printing
                         // for fast toggle
```

libsoc Example

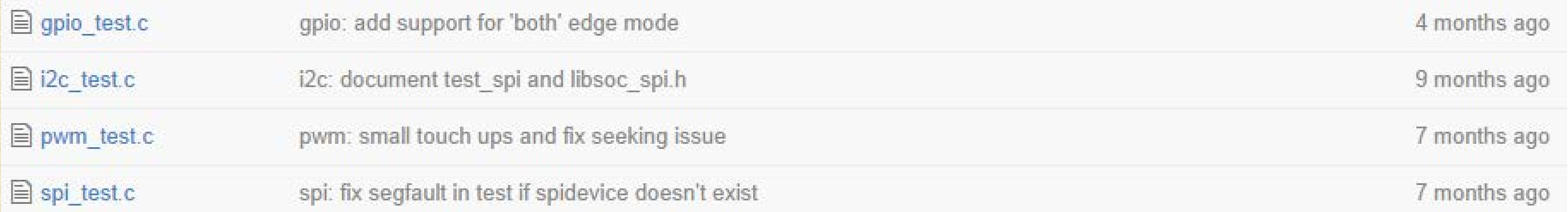
```
  int i;
  for (i=0; i<1000000; i++) { // Toggle GPIO X
times
    libsoc_gpio_set_level(gpio_output, HIGH);
    usleep(100000);        // sleep 100,000 uS
    libsoc_gpio_set_level(gpio_output, LOW);
    usleep(100000);
  }

  if (gpio_output) {
    // Free gpio request memory
    libsoc_gpio_free(gpio_output); }

  return EXIT_SUCCESS;
}
```

Libsoc - Compile and Run

```
bone$ gcc –lsoc -o blinkLED blinkLED.c
bone$ ./blinkLED
libsoc-debug: debug enabled (libsoc_set_debug)
libsoc-gpio-debug: requested gpio (50, libsoc_gpio_request)
libsoc-gpio-debug: GPIO already exported (50, libsoc_gpio_request)
libsoc-gpio-debug: setting direction to out (50, libsoc_gpio_set_direction)
libsoc-debug: debug disabled (libsoc_set_debug)
```

libsoc Examples

- https://github.com/jackmitch/libsoc/tree/master/test



gpio_test.c	gpio: add support for 'both' edge mode	4 months ago
i2c_test.c	i2c: document test_spi and libsoc_spi.h	9 months ago
pwm_test.c	pwm: small touch ups and fix seeking issue	7 months ago
spi_test.c	spi: fix segfault in test if spidevice doesn't exist	7 months ago

- exersizes/libsoc

Python GPIO

- Adafruit (http://www.adafruit.com/) has a nice BBB Python GPIO library
- https://github.com/adafruit/adafruit-beaglebone-io-python

Python install

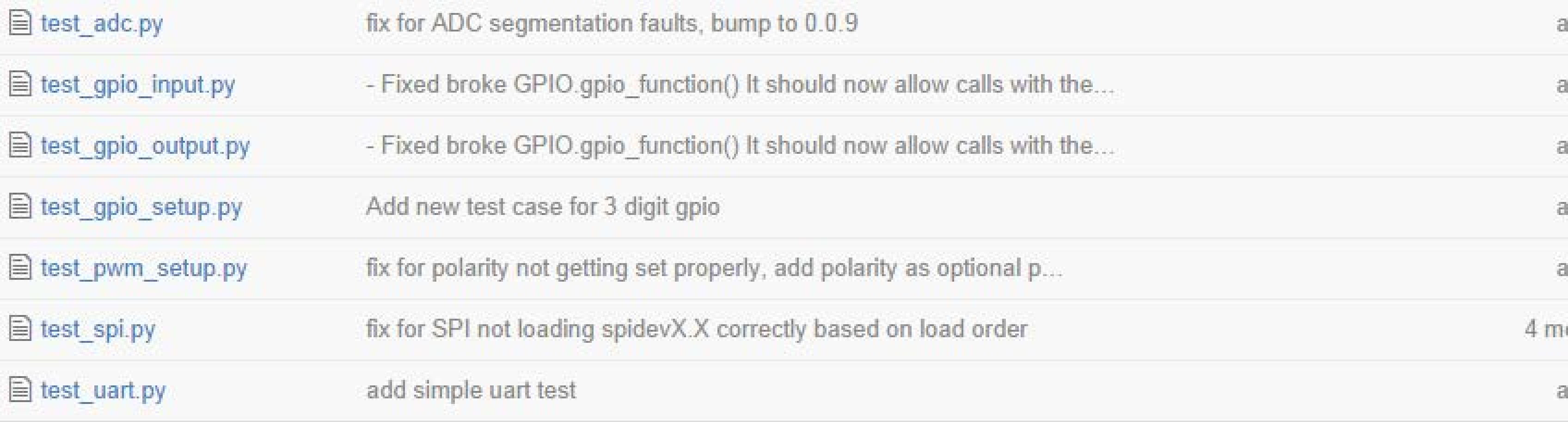
- Already installed

Python GPIO Code

```
#!/usr/bin/env python
import Adafruit_BBIO.GPIO as GPIO
import time

pin = "P9_14"

GPIO.setup(pin, GPIO.OUT)

while True:
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(pin, GPIO.LOW)
    time.sleep(0.5)
```

Python Examples

- https://github.com/adafruit/adafruit-beaglebone-io-python/tree/master/test



test_adc.py	fix for ADC segmentation faults, bump to 0.0.9	a
test_gpio_input.py	- Fixed broke GPIO.gpio_function() It should now allow calls with the...	a
test_gpio_output.py	- Fixed broke GPIO.gpio_function() It should now allow calls with the...	a
test_gpio_setup.py	Add new test case for 3 digit gpio	a
test_pwm_setup.py	fix for polarity not getting set properly, add polarity as optional p...	a
test_spi.py	fix for SPI not loading spidevX.X correctly based on load order	4 m
test_uart.py	add simple uart test	a

- exercises/python