



Getting Started with OSK's Pi-Sat

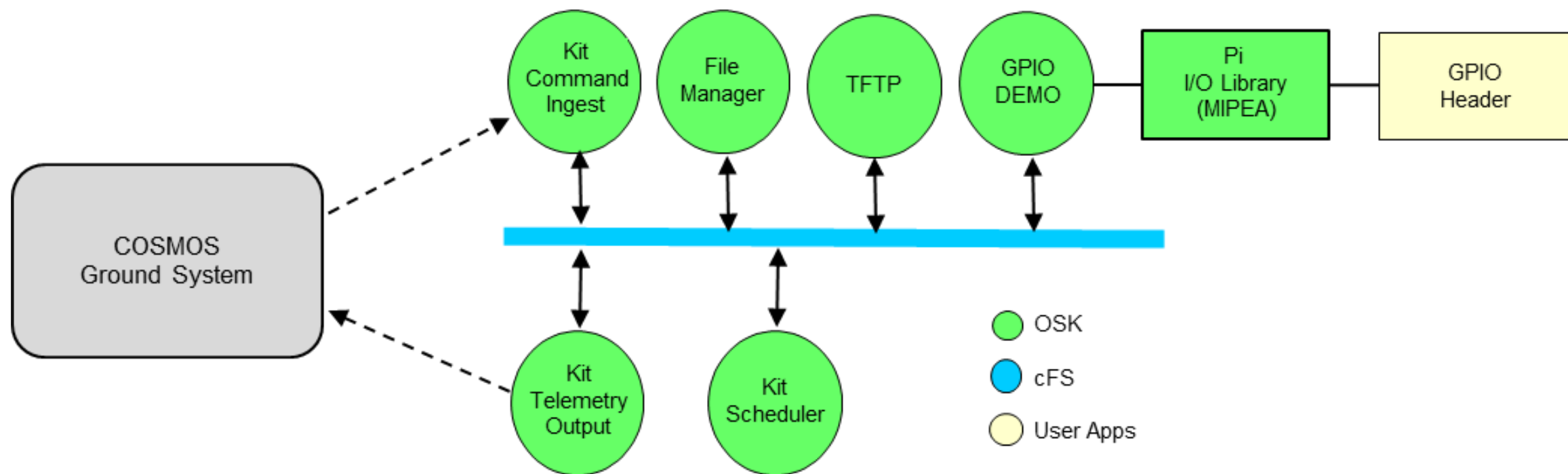
OSK v3.1
Pi-Sat v1.0



Introduction



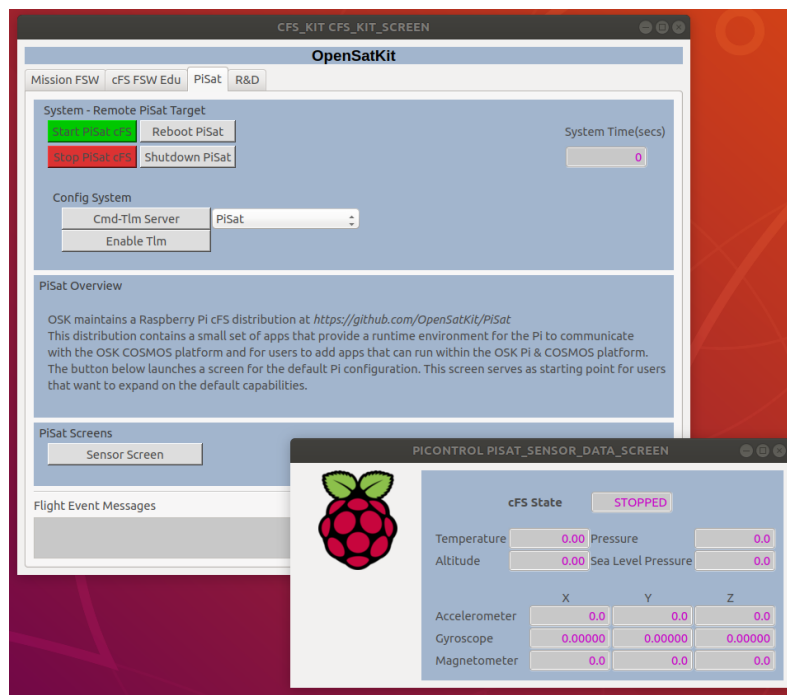
- **The primary objectives of the OpenSatKit (OSK) Pi-Sat are to**
 1. Provide a platform that allows STEM educators and hobbyists to work with NASA's core Flight System (cFS) on the low-cost Raspberry Pi
 2. Support a configuration that allows the COSMOS ground system to remotely communicate with the Pi-Sat over WiFi
 3. Provide pathways for user expansion and customizations
- **Rationale**
 - NASA's cFS is a mature flight-proven open-source flight software (FSW) framework that has many educational benefits
 - Remotely controlling Pi-Sat instantiations allows a user to gain an appreciation of spacecraft FSW development and operations
- **OSK does not provide a turn-key project**, so the documentation provides workflows and sequences of activities, but the details may vary based on user platforms
 - Users can create a "Pi-Sat Kit" that provides a turn-key solution for a particular situation
- **Perquisites**
 - Installation and running demo requires basic Linux operating system knowledge
 - Developing apps requires cFS runtime environment app group knowledge and C programming proficiency



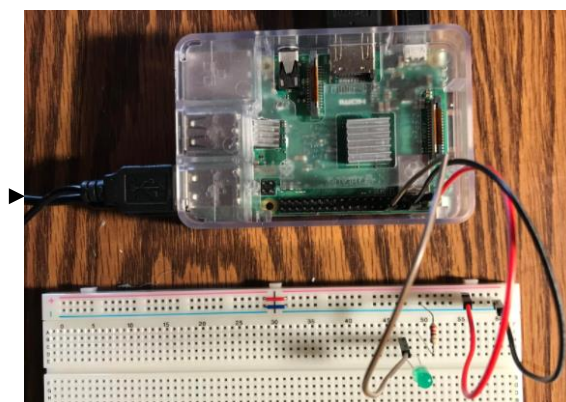
- Minimal set of OSK apps that provide an app runtime environment and file management/transfer services
- Only use OSK apps that use JSON tables and not cFS binary tables
- GPIO demo app can be used to verify the cFS Pi-Sat target installation and it can serve as a starting point for users to create their own apps
- Separate repo at <https://github.com/OpenSatKit/pi-sat>

1. **Configure your Pi hardware for GPIO demo**
 - Recommended so you can run the demo to verify later steps
2. **Install cFS Pi-Sat target on your Pi**
 - Run demo locally
3. **Configure your Pi as a WiFi access point**
4. **Configure COSMOS for Pi-Sat Communications**
5. **Connect COSMOS to PiSat**
 - Run demo remotely

COSMOS



Raspberry Pi GPIO Demo



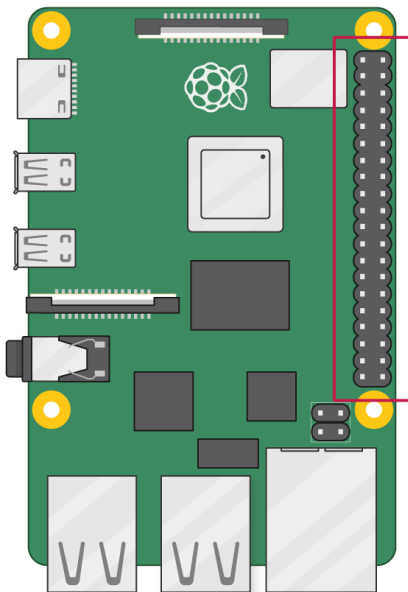
WiFi



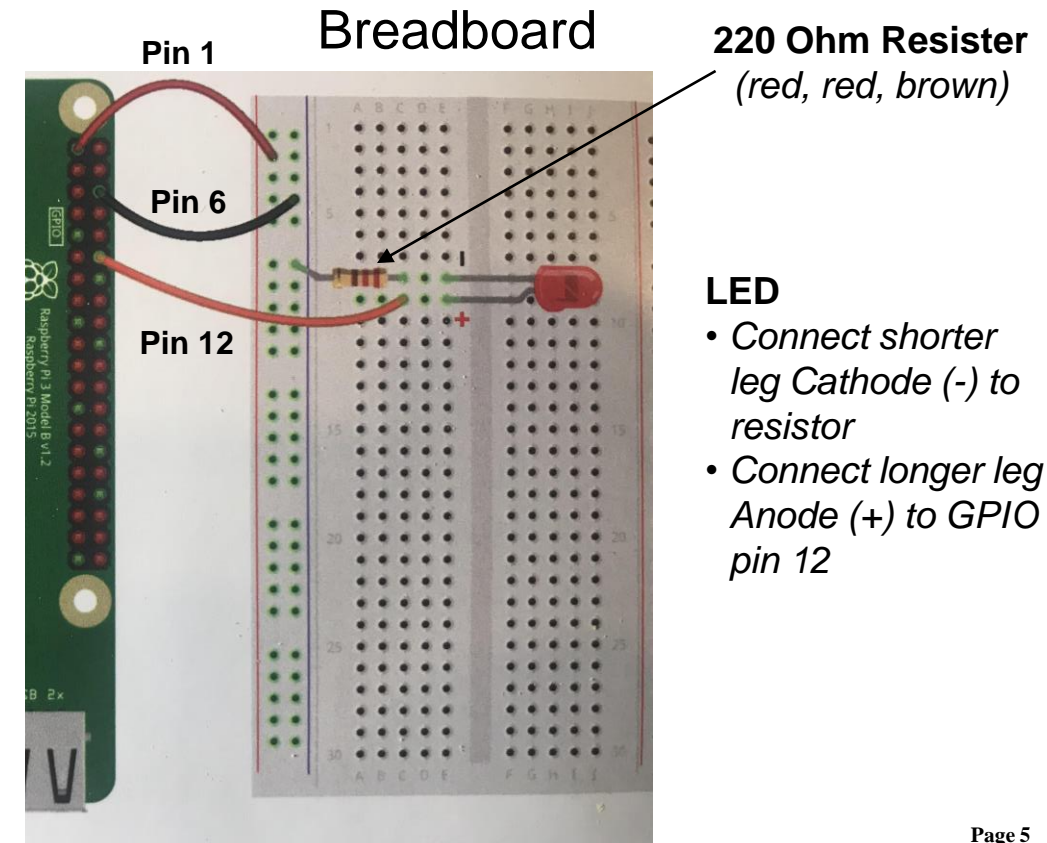
1 – Configure Your Pi Hardware for GPIO Demo

- The cFS Pi-Sat target comes with an app called **GPIO_DEMO** that blinks an LED connected to the Pi's General-Purpose Input/Output (GPIO)
 - Refer to the cFS Pi-Sat target section later in this package for software details
- **Configure and connect a breadboard to the Pi as shown below**
 - Note physical pin 12 is GPIO logical pin 18

40-Pin GPIO Header



3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)





2 – Install cFS Pi-Sat Target on your Pi



- **Prepare the development environment**

1. `sudo apt install cmake`

- **Clone OSK pi-sat repo and build the cFS target****

1. `git clone https://github.com/OpenSatKit/pi-sat`

2. Edit `cfs/apps/mipea/config.h` and define the Broadcom chip for your Pi

3. `cd pi-sat/cfs`

4. `make SIMULATION=native prep`

5. `make install`

- **Run the cFS target (must be run from executable location)**

1. `cd build/exe/cpu1`

2. `sudo ./core-cpu1`

- **GPIO Demo runs by default**

- It sends an information event message each time it turns on/off the LED

- If the LED is not blinking, then double check your LED wiring and the `mipea/config.h` setting

**Default configuration is for 32-bit Raspbian. See `/cfs/rpi_defs/toolchain-cpu1.cmake`



3 –Configure your Pi as a WiFi Access Point



- Coming in OSK v3.2

1. TBD



4 –Configure COSMOS for PiSat Communications



- Coming in OSK v3.2

1. TBD



5 – Connect COSMOS to Pi-Sat



- Coming in OSK v3.2

1. TBD



GPIO_DEMO App Design



GPIO_DEMO App Overview



- Describe GPIO app design so user's can use it as a starting point for their own designs