OptoSensor Reference

Firmware Reference

The Firmware structure can be divided into 4 parts Setup, Loop, Read ADC, and Output. The following is a detailed reference of the firmware and its functions.

setup()

The setup() function sets up all project variables and initializes both SPI communication interface and USB over TTL Serial communication channel, as well as sets up the firmware menu.

read_adc()

The read_adc() function polls the adcs for sensor values the function receives an int channel variable where the value of channel is between 1 – 8. The SPI Bus is configured in such a way that all adcs are triggered and channel ‘x’ is selected on all slaves the corresponding sensor values are then retrieved and placed within adcvalue[] array. i.e. if channel 1 is selected all adcs then output channel 1 values in a single call to this function.*

loop()

The loop() function loops through the menu cycle and remains within the menu until an is item selected.

raw()

raw() function calls the read_adc() function and outputs raw data from sensors without active filtering.

active()

active() function calls the read_adc() function and outputs active sensor data from sensors by filtering out sensor data below the threshold value. By default the threshold value is set to zero. Threshold value can be changed by either manual input of threshold level or by doing active threshold calculations which use avg. bg subtraction.

set_athreshold()

set_athreshold() function processes all sensors by doing a single pass through the sensors and then calculates the average sensor noise level, which it then uses to set the active threshold value.

set_threshold()

set_threshold() function sets threshold variable to what the user defines. Since the adc’s output a 12bit value in for the sensors i.e. 0 – 4095, the user can set a threshold between 0 – 4095

test_sensor()

The test_sensor() function polls a single user specified sensor and returns its value, used for debugging hardware.

s_frame()

The s_frame() function preforms a single pass to poll all sensors, and outputs data out within terminal terminal.

debug()

debug() function toggles the debug flag which displays contextual information in regards to sensor, and channel i.e. verbose output.