

# Pmod JSTK2 Reference Manual

The Diligent Pmod JSTK2 (Revision C) is a versatile user input device that can be easily incorporated into a wide variety of projects. With a two-axis joystick on a center button, a trigger button, and a programmable RGB LED capable of 24-bit color.





## Additional Important Documentation:

EMC Disclaimer, Digilent Development and Evaluation Kits ([https://digilent.com/reference/lib/exe/fetch.php?tok=a2aed5&media=https%3A%2F%2Ffiles.digilent.com%2Fresources%2Fdocuments%2FEMC\\_DISCLAIMER\\_DIGILENT\\_DEVELOPMENT\\_AND\\_EVALUATION\\_KITS.pdf](https://digilent.com/reference/lib/exe/fetch.php?tok=a2aed5&media=https%3A%2F%2Ffiles.digilent.com%2Fresources%2Fdocuments%2FEMC_DISCLAIMER_DIGILENT_DEVELOPMENT_AND_EVALUATION_KITS.pdf))

## Download 3D Print Files

- 3D print files for dual joystick ([https://digilent.com/reference/\\_media/reference/pmod/pmodjstk2/dual\\_joystick.zip](https://digilent.com/reference/_media/reference/pmod/pmodjstk2/dual_joystick.zip))
- 3D print files for single joystick ([https://digilent.com/reference/\\_media/reference/pmod/pmodjstk2/single\\_joystick.zip](https://digilent.com/reference/_media/reference/pmod/pmodjstk2/single_joystick.zip))

## Features

- Factory Calibrated Two Axis Resistive Joystick
- Center Joystick Button
- Trigger Style Push Button
- 24-bit RGB LED
- 6-pin Pmod connector with SPI interface
- Library and example code available in resource center

## Functional Description

The Pmod JSTK2 utilizes two potentiometers oriented orthogonally to one another and are manipulated by moving the joystick in the X and Y directions. As the joystick moves, the voltage output at the sweep pin of each potentiometer changes and is measured by the 10-bit ADC present on the embedded PIC16F1618 microcontroller. The raw measured data is stored at a rate of 100 Hz as a 16-bit right-justified variable in RAM with the upper 6 bits masked with zeros.

Additionally, each successive measurement also produces two 8-bit values representative of the joysticks physical location with respect to each axis. Note that if inversion of either of the 8-bit position axis are set, the values will not change until the data has been re-collected by the PIC16 at the 100 Hz rate.

## Specifications

| Parameter                         | Min   | Typical <sup>1</sup> | Max | Units   |
|-----------------------------------|-------|----------------------|-----|---------|
| Recommended Operating Voltage     | 3.1   | 3.3                  | 3.5 | V       |
| Maximum Supply Voltage            | -     | -                    | 5.5 | V       |
| Power Supply Current <sup>2</sup> | -     | 4.85                 | -   | mA      |
| Power Supply Current <sup>3</sup> | -     | 17.6                 | -   | mA      |
| Parameter                         | Value |                      |     | Units   |
| Maximum Joystick Angle            | 25    |                      |     | Degrees |
| Communication Protocol            | SPI   |                      |     |         |

Note<sup>1</sup> - Data in the Typical Column uses  $V_{CC}$  at 3.3V unless otherwise noted

Note<sup>2</sup> - Normal operation with the RGB LED Off and no buttons pressed

Note<sup>3</sup> - Normal operation with the RGB LED set to white and both buttons pressed

## Interfacing with the Pmod

The Pmod JSTK2 communicates with the host board via the SPI protocol with SPI Mode 0, CS active low, a 1 MHz clock, and each data byte organized MSb first. With the Pmod JSTK2, there are two types of data packet protocols: the standard data packet of 5 bytes and an extended data packet with 6 or 7 bytes in total. With the standard 5 byte protocol, users may use the old code from the Pmod JSTK without any syntax errors. The 5 byte packet structure is provided in the image below:

|             | Byte 1          | Byte 2           | Byte 3          | Byte 4           | Byte 5         |
|-------------|-----------------|------------------|-----------------|------------------|----------------|
| <b>MOSI</b> | COMMAND / 0     | PARAM1 / DUMMY   | PARAM2 / DUMMY  | PARAM3 / DUMMY   | PARAM4 / DUMMY |
| <b>MISO</b> | smpX (Low Byte) | smpX (High Byte) | smpY (Low Byte) | smpY (High Byte) | fsButtons      |

|                  | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1   | Bit 0    |
|------------------|--------|-------|-------|-------|-------|-------|---------|----------|
| <b>fsButtons</b> | EXTPKT | 0     | 0     | 0     | 0     | 0     | TRIGGER | JOYSTICK |

**EXTPKT:** Extended Packet Status Bit

1 = additional data corresponding to the command byte is available and may be retrieved after this byte

0 = standard response packet, no additional data follows this byte

**TRIGGER:** Trigger Button Status Bit

1 = trigger button is currently pressed

0 = trigger button is not being pressed

**JOYSTICK:** Joystick Center Button Status Bit

1 = joystick center button is currently pressed

0 = joystick center button is not being pressed

As noted in the standard data packet structure, users may either send a zero and a series of 4 dummy bytes to receive the standard 5 bytes of data or they may send a single command byte with up to 4 parameters in the four following bytes to set the internal values such as the joystick calibration or on-board RGB LED. The two bytes of data for each axis represent the 10-bit value for that axis. For the x-axis, a 0 value corresponds to the axis being tilted fully to the left and a value of 1023 corresponds fully to the right. Similarly, the y-axis has a value of 0 when it is tilted all the way down and a value of 1023 when it is tilted all the way up.

The extended data protocol allows for additional data to be obtained from the device during a communication session after the standard 5 bytes of information such as normalized 8-bit positional data for each axis. Users may also obtain the current calibration values and the status of the module through this method.

|             | Byte 1          | Byte 2           | Byte 3          | Byte 4           | Byte 5    | Byte 6   |
|-------------|-----------------|------------------|-----------------|------------------|-----------|----------|
| <b>MOSI</b> | cmdGetStatus    | DUMMY            | DUMMY           | DUMMY            | DUMMY     | DUMMY    |
| <b>MISO</b> | smpX (Low Byte) | smpX (High Byte) | smpY (Low Byte) | smpY (High Byte) | fsButtons | fsStatus |

|                 | Bit 7       | Bit 6   | Bit 5   | Bit 4   | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------|-------------|---------|---------|---------|-------|-------|-------|-------|
| <b>fsStatus</b> | CALIBRATING | LASTCAL | LASTFWS | LASTFRS | -     | -     | INVX  | INVY  |
| Initial Value   | 0           | 0       | 0       | 1       | 0     | 0     | 0     | 0     |

Bit 7 **CALIBRATING:** Calibration Status Bit<sup>(1)</sup>  
 1 = calibration procedure is currently executing  
 0 = calibration is not taking place

Bit 6 **LASTCAL:** Calibration Success Bit<sup>(2)</sup>  
 1 = last calibration procedure was successful  
 0 = last calibration procedure failed

Bit 5 **LASTFWS:** Flash Write Success Bit<sup>(3)</sup>  
 1 = last flash write was successful  
 0 = last flash write failed

Bit 4 **LASTFRS:** Flash Read Success Bit<sup>(4)</sup>  
 1 = last flash read was successful  
 0 = last flash read failed

Bit 3 **Unimplemented:** Read as '0'

Bit 2 **Unimplemented:** Read as '0'

Bit 1 **INVX:** Joystick X-Axis Position Inversion Enable Bit  
 1 = joystick x-axis position inversion is enabled  
 0 = joystick x-axis position inversion is disabled

Bit 0 **INVY:** Joystick Y-Axis Position Inversion Enable Bit  
 1 = joystick y-axis position inversion is enabled  
 0 = joystick y-axis position inversion is disabled

#### Note

1: This bit is set immediately after receiving a Calibrate command. It will remain set until the calibration procedure completes.

2: This bit is set or cleared immediately after a calibration procedure completes.

3: This bit is set or cleared immediately after a flash write attempt is performed. It will always be cleared at initial power on.

4: This bit is set or cleared immediately after a flash read attempt is performed. This bit will be set at initial power on, provided that the calibration constants were successfully read from the high endurance flash.

## SPI Timing Requirements

The embedded PIC16F1618 requires certain SPI timing requirements in order for successful communication to occur. When the Chip Select line is brought low, users must wait at least 15  $\mu$ s before sending the first byte of data. An interbyte delay of at least 10  $\mu$ s is required when transferring multiple bytes. When the Chip Select line is brought high after the last byte has been transferred, at least 25  $\mu$ s is required before users may bring the Chip Select line low again to initiate another communication session.

## Calibrating the Module

The Pmod JSTK2 has a set of factory loaded calibration values that are used to calculate the 8-bit position values for each axis. Users may enter calibration mode to recalculate all of those values by rotating the joystick around so the embedded PIC16 can record all of the maximum and minimum samples for the two axes. The on-board blue LED will be flashing to indicate that the calibration sequence is taking place. When the embedded microcontroller detects that the joystick has not changed for an entire second, allowing the microcontroller to presume that the most recent set of measurements correspond to the joystick's center position, the blue LED will stop flashing and the green LED will flash twice to indicate that the calibration procedure was successful. However, if 10 seconds pass without the PIC16 detecting the center position, the blue LED will stop flashing and the red LED will flash twice indicating that the calibration procedure was not successful.

Once the Chip Select pin goes high after the calibration command has been processed, the Pmod JSTK2 will not accept any new commands during the calibration procedure. Users may still poll the status register to determine the current status of the device during this time.

## Using the High Endurance Flash

The calibration constants that the module uses are loaded into RAM from the High Endurance Flash (HEF) during the firmware initialization which occurs shortly after power is supplied. If a calibration sequence is performed, new calibration constants are loaded into RAM. In general, it is not necessary to then write the new calibration constants in RAM to the HEF unless the user wishes to have those new calibration constants available after a power cycle. The factory loaded calibration constants are available in the header file for the Pmod JSTK2 in the Example Code ([https://digilent.com/reference/pmod/pmodjstk2/start#example\\_projects](https://digilent.com/reference/pmod/pmodjstk2/start#example_projects)).

If the user does want to store the calibration constants in a more permanent fashion, they may issue a command to write the constants from RAM to Flash, but it is recommended to wait at least 5 milliseconds once the Chip Select goes high after the data transfer is complete because the device is unable to respond to traffic on the SPI bus.

Users may also choose to reload the constants from HEF back into RAM which will take up to 95 microseconds to complete, so it is recommend to wait at least 100 microseconds after the Chip Select is pulled high after the data transfer. Much like the RAM to Flash command, the module will not be able to respond to SPI commands during this time.

---

## Command Summary

A list of commands supported by the Pmod JSTK2 library code ([https://digilent.com/reference/pmod/pmodjstk2/start#example\\_projects](https://digilent.com/reference/pmod/pmodjstk2/start#example_projects)) is provided below.

## Get Commands

|                       |               |       |       |       |       |       |       |       |       |
|-----------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                       |               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| <b>cmdGetPosition</b> | <b>(0xC0)</b> | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| <i>Parameters</i>     |               |       |       |       |       |       |       |       |       |
| None                  |               |       |       |       |       |       |       |       |       |

Get the 8-bit position variables corresponding to the location of the Joystick's X and Y axis. The X position is transferred to the master following the byte containing the button state. The Y position is transferred to the master immediately following the X position.

|                     |               |       |       |       |       |       |       |       |       |
|---------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                     |               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| <b>cmdGetStatus</b> | <b>(0xF0)</b> | 1     | 1     | 1     | 1     | 0     | 0     | 0     | 0     |
| <i>Parameters</i>   |               |       |       |       |       |       |       |       |       |
| None                |               |       |       |       |       |       |       |       |       |

Get a copy of the device's status register. The 8-bit status register is transferred to the master following the byte containing the button state.

|                          |               |       |       |       |       |       |       |       |       |
|--------------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                          |               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| <b>cmdGetFirmwareVer</b> | <b>(0xF1)</b> | 1     | 1     | 1     | 1     | 0     | 0     | 0     | 1     |
| <i>Parameters</i>        |               |       |       |       |       |       |       |       |       |
| None                     |               |       |       |       |       |       |       |       |       |

Get a copy of the device's firmware version. The low byte of the firmware version is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                      |               |       |       |       |       |       |       |       |       |
|----------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                      |               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| <b>cmdGetCalXMin</b> | <b>(0xE0)</b> | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     |
| <i>Parameters</i>    |               |       |       |       |       |       |       |       |       |
| None                 |               |       |       |       |       |       |       |       |       |

Get a copy of the smpXMin calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                      |               |       |       |       |       |       |       |       |       |
|----------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
|                      |               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| <b>cmdGetCalXMax</b> | <b>(0xE1)</b> | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 1     |
| <i>Parameters</i>    |               |       |       |       |       |       |       |       |       |
| None                 |               |       |       |       |       |       |       |       |       |

Get a copy of the smpXMax calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalYMin</b> (0xE2) | 1     | 1     | 1     | 0     | 0     | 0     | 1     | 0     |

*Parameters*

None

Get a copy of the smpYMin calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalYMax</b> (0xE3) | 1     | 1     | 1     | 0     | 0     | 0     | 1     | 1     |

*Parameters*

None

Get a copy of the smpYMax calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalXCenMin</b> (0xE4) | 1     | 1     | 1     | 0     | 0     | 1     | 0     | 0     |

*Parameters*

None

Get a copy of the smpXCenterMin calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalXCenMax</b> (0xE5) | 1     | 1     | 1     | 0     | 0     | 1     | 0     | 1     |

*Parameters*

None

Get a copy of the smpXCenterMax calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalYCenMin</b> (0xE6) | 1     | 1     | 1     | 0     | 0     | 1     | 1     | 0     |

*Parameters*

None

Get a copy of the smpYCenterMin calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdGetCalYCenMax</b> (0xE7) | 1     | 1     | 1     | 0     | 0     | 1     | 1     | 1     |

*Parameters*

None

Get a copy of the smpYCenterMax calibration constant. The low byte of the calibration constant is transferred to the master following the byte containing the button state. The high byte is transferred to the master immediately following the low byte.

## Set Commands

|                         | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1   | Bit 0 |
|-------------------------|-------|-------|-------|-------|-------|-------|---------|-------|
| <b>cmdSetLed</b> (0x80) | 1     | 0     | 0     | 0     | 0     | 0     | IGNORED | LEDST |

*Parameters*  
None

**LEDST:** Green LED ON/OFF State  
1 = turn LED on  
0 = turn LED off

**IGNORED:** the state of this bit has ignored

Turn the Green LED on or off. The Red and Green LEDs are both set to the off state.

|                            | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetLedRGB</b> (0x84) | 1     | 0     | 0     | 0     | 0     | 1     | 0     | 0     |

*Parameters*  
PARAM1 – Red LED duty cycle  
PARAM2 – Green LED duty cycle  
PARAM3 – Blue LED duty cycle  
PARAM4 – ignored

Set the duty cycles for the Red, Green, and Blue LEDs.

|                               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetInversion</b> (0x88) | 1     | 0     | 0     | 0     | 1     | 0     | INVX  | INVY  |

*Parameters*  
None

**INVX:** Joystick X-Axis Position Inversion Enable Bit  
1 = enable joystick x-axis position inversion  
0 = disable joystick x-axis position inversion

**INVY:** Joystick Y-Axis Position Inversion Enable Bit  
1 = enable joystick y-axis position inversion  
0 = disable joystick y-axis position inversion

Enable inversion of the 8-bit position value corresponding to the X and/or Y axis of the Joystick.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXMin</b> (0xA8) | 1     | 0     | 1     | 0     | 1     | 0     | 0     | 0     |

*Parameters*  
PARAM1 – smpXMin (Low Byte)  
PARAM2 – smpXMin (High Byte)  
PARAM3 – ignored  
PARAM4 – ignored

Set the smpXMin calibration constant. Please note that the value specified for smpXMin must be less than smpXCenterMin. If the specified value fails to meet this requirement then smpXMin will not be updated.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXMax</b> (0xA9) | 1     | 0     | 1     | 0     | 1     | 0     | 0     | 1     |

*Parameters*  
PARAM1 – smpXMax (Low Byte)  
PARAM2 – smpXMax (High Byte)  
PARAM3 – ignored  
PARAM4 – ignored

Set the smpXMax calibration constant. Please note that the value specified for smpXMax must be greater than smpXCenterMax. If the specified value fails to meet this requirement then smpXMax will not be updated.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYMin</b> (0xAA) | 1     | 0     | 1     | 0     | 1     | 0     | 1     | 0     |

*Parameters*  
PARAM1 – smpYMin (Low Byte)  
PARAM2 – smpYMin (High Byte)  
PARAM3 – ignored  
PARAM4 – ignored

Set the smpYMin calibration constant. Please note that the value specified for smpYMin must be less than smpYCenterMin. If the specified value fails to meet this requirement then smpYMin will not be updated.

|                             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYMax</b> (0xAB) | 1     | 0     | 1     | 0     | 1     | 0     | 1     | 1     |

*Parameters*

- PARAM1 – smpYMax (Low Byte)
- PARAM2 – smpYMax (High Byte)
- PARAM3 – ignored
- PARAM4 – ignored

Set the smpYMax calibration constant. Please note that the value specified for smpYMax must be greater than smpYCenterMax. If the specified value fails to meet this requirement then smpYMax will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXCenMin</b> (0xAC) | 1     | 0     | 1     | 0     | 1     | 1     | 0     | 0     |

*Parameters*

- PARAM1 – smpXCenterMin (Low Byte)
- PARAM2 – smpXCenterMin (High Byte)
- PARAM3 – ignored
- PARAM4 – ignored

Set the smpXCenterMin calibration constant. Please note that the value specified for smpXCenterMin must be greater than smpXMin and less than smpXCenterMax. If the specified value fails to meet these requirements then smpXCenterMin will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXCenMax</b> (0xAD) | 1     | 0     | 1     | 0     | 1     | 1     | 0     | 1     |

*Parameters*

- PARAM1 – smpXCenterMax (Low Byte)
- PARAM2 – smpXCenterMax (High Byte)
- PARAM3 – ignored
- PARAM4 – ignored

Set the smpXCenterMax calibration constant. Please note that the value specified for smpXCenterMax must be greater than smpXCenterMin and less than smpXMax. If the specified value fails to meet these requirements then smpXCenterMax will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYcenMin</b> (0xAE) | 1     | 0     | 1     | 0     | 1     | 1     | 1     | 0     |

*Parameters*

- PARAM1 – smpYCenterMin (Low Byte)
- PARAM2 – smpYCenterMin (High Byte)
- PARAM3 – ignored
- PARAM4 – ignored

Set the smpYCenterMin calibration constant. Please note that the value specified for smpYCenterMin must be greater than smpYMin and less than smpYCenterMax. If the specified value fails to meet these requirements then smpYCenterMin will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYcenMax</b> (0xAF) | 1     | 0     | 1     | 0     | 1     | 1     | 1     | 1     |

*Parameters*

- PARAM1 – smpYCenterMax (Low Byte)
- PARAM2 – smpYCenterMax (High Byte)
- PARAM3 – ignored
- PARAM4 – ignored

Set the smpYCenterMax calibration constant. Please note that the value specified for smpYCenterMax must be greater than smpYCenterMin and less than smpYMax. If the specified value fails to meet these requirements then smpYCenterMax will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXMinMax (0xB0)</b> | 1     | 0     | 1     | 1     | 0     | 0     | 0     | 0     |

*Parameters*

- PARAM1 – smpXMin (Low Byte)
- PARAM2 – smpXMin (High Byte)
- PARAM3 – smpXMax (Low Byte)
- PARAM4 – smpXMax (High Byte)

Set the smpXMin and smpXMax calibration constants. Please note that the value specified for smpXMin must be less than smpXCenterMin and the value specified for smpXMax must be greater than smpXCenterMax. If either of the specified values fail to meet the above requirements then smpXMin and smpXMax will not be updated.

|                                | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYMinMax (0xB1)</b> | 1     | 0     | 1     | 1     | 0     | 0     | 0     | 1     |

*Parameters*

- PARAM1 – smpYMin (Low Byte)
- PARAM2 – smpYMin (High Byte)
- PARAM3 – smpYMax (Low Byte)
- PARAM4 – smpYMax (High Byte)

Set the smpYMin and smpYMax calibration constants. Please note that the value specified for smpYMin must be less than smpYCenterMin and the value specified for smpYMax must be greater than smpYCenterMax. If either of the specified values fail to meet the above requirements then smpYMin and smpYMax will not be updated.

|                                   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalXCenMinMax (0xB2)</b> | 1     | 0     | 1     | 1     | 0     | 0     | 1     | 0     |

*Parameters*

- PARAM1 – smpXCenterMin (Low Byte)
- PARAM2 – smpXCenterMin (High Byte)
- PARAM3 – smpXCenterMax (Low Byte)
- PARAM4 – smpXCenterMax (High Byte)

Set the smpXCenterMin and smpXCenterMax calibration constants. Please note that the value specified for smpXCenterMin must be greater than smpXMin and the value specified for smpXCenterMax must be less than smpXMax. Additionally, the value specified for smpXCenterMin must be less than the value specified for smpXCenterMax. If any of the specified values fail to meet the above requirements then smpXCenterMin and smpXCenterMax will not be updated.

|                                   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>cmdSetCalYCenMinMax (0xB3)</b> | 1     | 0     | 1     | 1     | 0     | 0     | 1     | 1     |

*Parameters*

- PARAM1 – smpYCenterMin (Low Byte)
- PARAM2 – smpYCenterMin (High Byte)
- PARAM3 – smpYCenterMax (Low Byte)
- PARAM4 – smpYCenterMax (High Byte)

Set the smpYCenterMin and smpYCenterMax calibration constants. Please note that the value specified for smpYCenterMin must be greater than smpYMin and the value specified for smpYCenterMax must be less than smpYMax. Additionally, the value specified for smpYCenterMin must be less than the value specified for smpYCenterMax. If any of the specified values fail to meet the above requirements then smpYCenterMin and smpYCenterMax will not be updated.

## Other Commands

|  |        |       |       |       |       |       |       |       |   |
|--|--------|-------|-------|-------|-------|-------|-------|-------|---|
|  | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |   |
| <b>cmdCalibrate</b><br><i>Parameters</i><br>None<br>Enter Joystick calibration mode. | (0xA4) | 1     | 0     | 1     | 0     | 0     | 1     | 0     | 0 |

|  |        |       |       |       |       |       |       |       |   |
|--|--------|-------|-------|-------|-------|-------|-------|-------|---|
|  | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |   |
| <b>cmdWriteFlash</b><br><i>Parameters</i><br>None<br>Write the calibration constants from RAM to High Endurance Flash. | (0xB8) | 1     | 0     | 1     | 1     | 1     | 0     | 0     | 0 |

|   |        |       |       |       |       |       |       |       |   |
|---|--------|-------|-------|-------|-------|-------|-------|-------|---|
|   | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |   |
| <b>cmdRidFromFlash</b><br><i>Parameters</i><br>None<br>Read the calibration constants from High Endurance Flash and load them into RAM. | (0xBC) | 1     | 0     | 1     | 1     | 1     | 1     | 0     | 0 |

## Pinout Description Table

A pinout table of the Pmod JSTK2 is provided below.

| Header J1 |        |                     |
|-----------|--------|---------------------|
| Pin       | Signal | Description         |
| 1         | CS     | Chip Select         |
| 2         | MOSI   | Master-Out-Slave-In |
| 3         | MISO   | Master-In-Slave-Out |
| 4         | SCK    | Serial Clock        |
| 5         | GND    | Power Supply Ground |
| 6         | VCC    | Power Supply (3.3V) |


Although users are welcome to create their own interface code for the Pmod JSTK2 if they so desire, although pre-constructed libraries that provide functions for initializing the module, reading in values, and adjusting calibration values exist. They are available on the Pmod JSTK2 Example Code section ([https://digilent.com/reference/pmod/pmodjstk2/start#example\\_projects](https://digilent.com/reference/pmod/pmodjstk2/start#example_projects)).

Any external power applied to the Pmod JSTK2 must be within 2.95V and 5.5V; however, it is recommended that Pmod is operated at 3.3V.


## Physical Dimensions

The pins on the pin header are spaced 100 mil apart. The PCB is 1.875 inches long on the sides parallel to the pins on the pin header, 0.9375 inch long on the sides perpendicular to the pin header, and 1.75 inches tall. With the 3-D printed housing the module is 1.875 inches long on the sides parallel to the pins on the pin header, 1.125 inches long on the sides perpendicular to the pin header, and 1.75 inches tall.

## Additional Information

Additional information about the PIC16F1618 including communication modes and specific timings of the chip can be found by checking out its datasheet  here (<http://ww1.microchip.com/downloads/en/DeviceDoc/40001769B.pdf>).

Example code demonstrating how to get information from the Pmod JSTK2 can be found here ([https://digilent.com/reference/pmod/pmodjstk2/start#example\\_projects](https://digilent.com/reference/pmod/pmodjstk2/start#example_projects)).

If you have any questions or comments about the Pmod JSTK2, feel free to post them under the appropriate section (“Add-on Boards”) of the  Digilent Forum (<https://forum.digilent.com/>).