

USB-GPIO12 General Purpose PIC18 Processor Module

Technical Specification



Overview

The USB-GPIO12 is a general purpose processor module using the PIC18F14K50. This chip has a USB module built in which we have used to implement Microchip's HID Boot-loader. This boot-loader is already programmed into USB-GPIO12 and ready for use. The USB-GPIO12 is powered from the USB bus and the 5v supply is available for your own circuits. Maximum current required by the module is 20mA with both LED's on, leaving 80mA available to your circuits from a standard 100mA USB port.

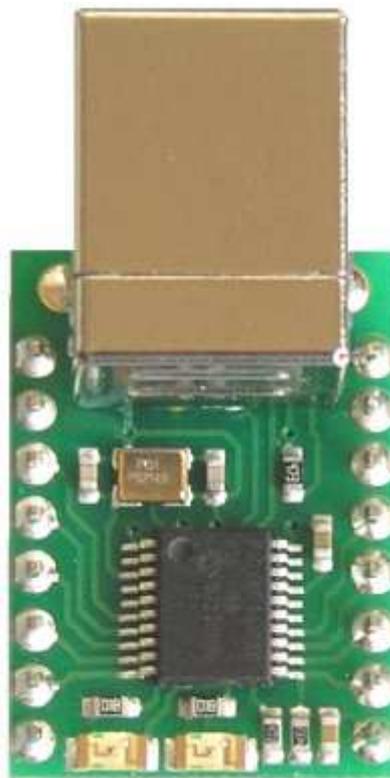
Schematic

The schematic of the GPIO12 is here [usb-gpio12.pdf](#)

Connections

The diagram below shows the connections for the various modes.

Pull low at power-up for boot-loader mode, RA3 - 1
Red Led, CCP1/RC5 - 2
Green Led, RC4 - 3
RC3/AN7 - 4
RC6/AN8 - 5
RC7/AN9/SDO - 6
RB7/Tx - 7
0v Ground - 8



16 - 5v Power from USB Cable
15 - RC0/AN4
14 - RC1/AN5
13 - RC2/AN6
12 - RB4/AN10/SDI/SDA
11 - RB5/AN11/Rx
10 - RB6/SCK/SCL
9 - 0v Ground

I/O pins

The USB-GPIO12 has 12 I/O pins, 13 if you count the input only RA3 pin.

You should consult the Microchip data sheet for the PIC18F14K50 for full details and programming information. We have included 4k7 pull-up resistors on pins 10 & 12 (SCL and SDA) to make I2C interfacing easier.

Software Requirements

To use USB-GPIO12 you will need at least the following:
MPLAB Microchips IDE,

MCC18 compiler,
Microchip Application Libraries.

If you don't have them, you should go to www.microchip.com and download and install them now.

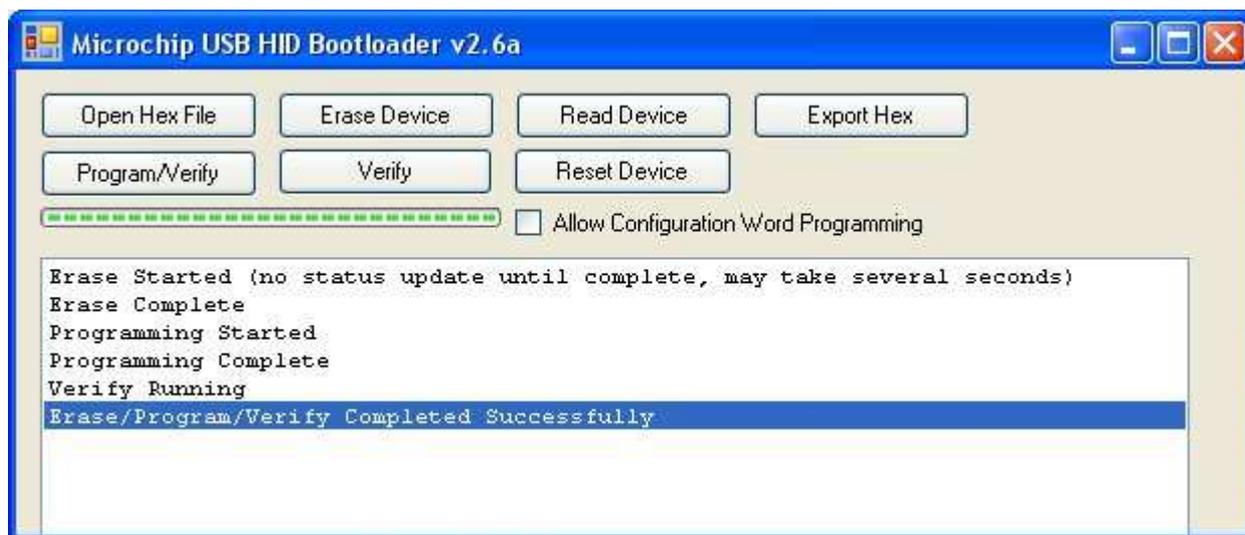
Boot Loader

The built in boot loader is the one provided by Microchip as part of the Application Libraries. You will need to download the Libraries as they contain "HIDBootLoader.exe". This is the PC side driver program which downloads your software to the USB-GPIO12. If you installed the libraries in the default position it will be here:
C:\Microchip Solutions\USB Device - Bootloaders\HID - Bootloader\HIDBootLoader.exe

When you first power up the the Green LED will be flashing and all other pins will be high impedance. This is a very simple application program we load after testing the module in our workshop. To download you own program you must first put the USB-GPIO12 into boot-loader mode. Do this by connecting pin 1 to 0v Ground, then connect the USB cable to power the module. The Red LED will light and the Green LED will be off. Now run HIDBootLoader.exe and you should see:



Open the hex file for your project and press Program/Verify. This will down load your program into the module.



Now remove the link between pin1 and 0v ground and press "Reset Device". This will reset the module and run your application.

You can always get back to the boot-loader by by connecting pin 1 to 0v ground and cycling the power, but there is a better way. It relies on you placing code in your program to perform a software reset when pin RA3 goes low. To do this effectively you need to check the pin at least every few mS. As an example load up the "shipping" project (See examples below) and look at line 122:

```
while(!PIR2bits.TMR3IF; // wait 1mS
```

This is part of the 1mS delay routine and is a good place to add our check. Change the line to:

while(!PIR2bits.TMR3IF) if(!PORTAbits.RA3) Reset(); // wait 1mS, going to the boot loader if RA3 is pulled low
If you wire a small push button switch between pin1 and 0v ground you can now get back to the boot loader with a button press. If you still have HIDBootLoader.exe running it will automatically reconnect ready to download a new program.

Examples

We have some [starter projects](#) for you to try.

Shipping is the simple Green LED flashing program that was in the module when you received it.

PinTest is a program which sets one pin low at a time and cycles through all 12 pins every 2 seconds. Wiring an LED via a 390 ohm resistor to 5v will make a nice test for the I/O lines.

ComPortEcho is a USB com port example (from the Microchip Application Library). It sets the module as a com port and echoes back each character you send it incremented by 1. 'A' comes back as 'B'.

Compiling programs to run with the boot-loader

The boot loader occupies program memory from 0x0000 to 0x0FFF. Your programs must therefore be linked to run at 0x1000. The reset and interrupt vectors which are normally at 0x0000, 0x0008 and 0x0018 are re-mapped to 0x1000, 0x1008 and 0x1018. The example programs above show how to do this. You should study those for the details, or just copy them as a starting point for your own projects. You must also include the rm18F14K50.lkr linker script in your project. This tells the linker how to re-locate your program to the correct place.