GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

**ECE 4550 — Control System Design — Summer 2018**

# F28027 Launch Pad Information

## Contents

## 1 Introduction

The purpose of this document is to answer a few questions that may arise in connection with your use of the F28027 Launch Pad. This kit is described elsewhere (e.g. in its User's Guide), so there is no need to simply re-state information that is best extracted from fundamental sources; instead, the intent is to provide supplementary explanation of issues that students may not fully grasp after consulting the fundamental sources.

## 2 Power Isolation

### 2.1 Two Power Domains

One feature that separates the F28027 Launch Pad from some other TI Launch Pads is its use of isolated power domains. This feature is most easily visualized by consulting the printed circuit board layout displayed in Figure 4 (top copper layer) and Figure 5 (bottom copper layer) of the User's Guide. In these figures, there are two distinct regions in which the copper layers are used to interconnect components, and the two regions are separated by copper-free voids (seen as vertical white strips). The larger of these two regions is where the microcontroller is located along with header pins, configuration switches, push buttons and a row of LEDs; the smaller of these two regions is where the JTAG emulator circuitry is located along with a USB connector.

Figure 2 of the User's Guide further clarifies the role of the isolated power domains. To the left of that figure are the USB connector, integrated circuit U4 which develops a 3.3 V supply from the 5 V pin of the USB connector, and integrated circuits U6 and U8 which together provide JTAG emulation capability. To the right of that figure is the isolation barrier between the two power domains; integrated circuits U5 and U7 permit digital signals to pass between the two power domains by crossing the isolation barrier. These digital signals fall into two categories.

1. JTAG emulation signals (see Section 4.8 of the F28027 Datasheet)
   (a) Microcontroller input signals: JTAG_TRST, TCK, TMS, TDI
   (b) Microcontroller output signal: TDO
2. UART serial communication signals (see Section 4.3 of the F28027 Datasheet)
   (a) Microcontroller input signal: SCI_RX
   (b) Microcontroller output signal: SCI_TX

Why go to the trouble of adding an isolation barrier in the first place? Among the motivations are human safety, equipment protection, and noise insensitivity, all of which are relevant in applications involving power electronics and electric machines. Signaling between the microcontroller (which is presumably embedded within a system characterized by appreciable power flow) and a host computer is desired, but without direct electrical connections between their circuitry.

The User's Guide is a bit confusing as Figure 2 shows a host of 0 Ω resistors (R16, R18, R19, R20, R21, R25, R28, R30 and R32) defeating the isolation barrier, and these resistors are even listed in the Bill of Materials (Table 2). In reality, none of these short-circuiting resistors has been placed on the board, though the pads for these resistors are clearly visible in the top copper layer of Figure 4. The only possible way to install these resistors so as to defeat the isolation barrier would be to first remove integrated circuits U5 and U7 (the resistor pads are located directly beneath these chips on the top copper layer), so it seems that this feature was added to the board by the board designer during its development only for initial hardware debugging purposes.

## 2.2 Configuration Options

Most relevant for our purposes are the three pairs of header pins used to configure the system for operation with *isolated* power domains (jumpers omitted) or *connected* power domains (jumpers installed); JP1 is for the 3.3 V bus, JP2 is for the ground bus and JP3 is for the 5 V bus. Jumper JP3 is typically unnecessary; if installed, it would provide 5 V to a connected (but not externally powered) Booster Pack[1] at header pin J5-1 through the USB cable. Typical power domain configuration options are described below.

1. If you want the microcontroller to be powered from the USB cable, then you need to install jumpers JP1 and JP2. Power domain isolation will be defeated, but this is not a problem provided that there is no high-power-capable Booster Pack connected to the Launch Pad.
2. If the USB cable is removed for embedded deployment and you intend to provide microcontroller power through header J3 without powering the (unused) circuitry that enables communication over the USB cable, then you need to remove jumpers JP1 and JP2.
3. If you are working with a connected, externally powered, high-power-capable Booster Pack and you need to safely communicate with the microcontroller using the USB cable, then jumpers JP1, JP2 and JP3 should be removed to achieve full power domain isolation.

[1] A Booster Pack is a circuit board that mechanically and electrically connects to the Launch Pad circuit board via stackable headers J1, J2, J5 and J6. Some Booster Packs involve only "information processing" circuitry, but others involve "power processing" circuitry and hence benefit from power isolation. For example, if a high-power-capable motor driver Booster Pack is connected to the Launch Pad, then power isolation would be appropriate.

# 3 Configuration Switches

## 3.1 Boot Mode Options

The microcontroller contains a factory-programmed internal ROM boot-loader, as described in Section 3.2.8 of the F28027 Datasheet. Complete documentation of the boot process appears in the F28027 Boot ROM Reference Guide. The boot-loader uses the state of the TRST pin and two GPIO pins (GPIO34 and GPIO37) to determine which boot mode to use. All three of these pin states may be assigned using the boot mode switch on the Launch Pad board; this switch appears as S1 in Figure 3 of the User's Guide, and it is labeled as "Boot" on the board's silk screen. For our purposes, we need only the following two standard boot modes:

1. Emulation Boot to RAM: If Boot-3 is in the up/on position, then the microcontroller will perform an emulation boot to the program stored in its internal RAM memory, irrespective of the settings on Boot-1 and Boot-2. This is the boot mode we want during any CCS debug session (including one used only to program the internal flash memory).

2. Stand-Alone Boot to Flash: If Boot-3 is in the down/off position and both Boot-1 and Boot-2 are in the up/on position, then the microcontroller will perform a stand-alone boot to the program stored in its internal flash memory. This is the boot mode that we typically want when designing an embedded stand-alone application.

## 3.2 Serial Communication Options

Common system-to-system serial communication options include Ethernet, USB, CAN and UART. Some microcontrollers offer Ethernet, USB or CAN peripheral modules, but the F28027 includes only the UART option through its SCI peripheral module. Launch Pad users may wish to utilize this serial communication capability in stand-alone embedded applications, and the board design makes this possible in the following two different modes (enabled by the switch labeled S4 on Figure 3 of the User's Guide and labeled "Serial" on the board's silk screen):

1. If the microcontroller SCI peripheral is used to communicate with another system (e.g. a computer) through the USB connector, then Serial should be set to the up/on position.

2. If the microcontroller SCI peripheral is used to communicate with a Booster Pack board through the stackable header pins, then Serial should be set to the down/off position.

# 4 Header Pins

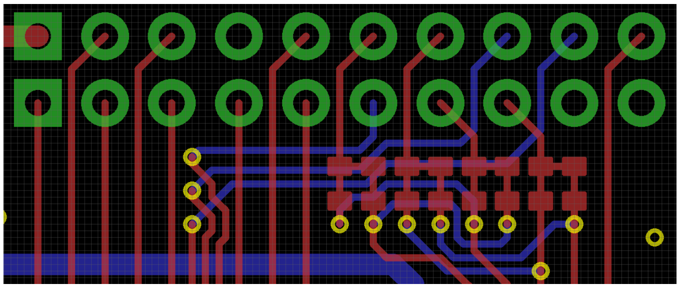
## 4.1 ADC Header Pins

If you were to use the F28027 Launch Pad in projects outside of this course, then you may need to know about a feature of F28027 microcontrollers (actually the entire Piccolo family, including F28069) not specifically considered in any of our regular labs. Microcontrollers having a moderate number of pins (e.g. 48 instead of 176) are desirable because they often cost less and have smaller footprints. As we have seen, GPIO pin multiplexing enhances design flexibility by allowing each GPIO pin to be internally connected to one of several possible digital peripheral circuits. In a similar way, pin multiplexing may also be used to modify the function of ADC pins. According to Table 2-2 of the F28027 Datasheet, ADCINA2, ADCINA4, ADCINA6, ADCINB2, ADCINB4 and ADCINB6 may be configured as GPIO pins; this increases the number of available GPIO pins by six, which may be useful in some applications (all six pins are available on headers J1, J2 and J5).

## 4.2 GPIO Header Pins

A potentially confusing aspect of the board's circuit design is the way certain GPIO pins have been interconnected. According to Table 2-2 of the F28027 Datasheet, the 18 usable GPIO pins are labeled GPIO0-7, GPIO12, GPIO16-19, GPIO28-29 and GPIO32-34. A subset of 14 of these 18 usable GPIO pins has been made available in a standard fashion on headers J1, J2 and J6, but the remaining 4 of these 18 usable GPIO pins, specifically GPIO16-17 and GPIO32-33, are connected to headers J2 and J6 in a clever but non-intuitive way. The design approach taken for GPIO16-17 and GPIO32-33 was motivated by a desire to maximize compatibility with the header pin specification for the first TI Launch Pad boards (which were based on the MSP430 family of microcontrollers); TI has four separate categories of Launch Pad boards[2] so definition of a common header pin specification improves the chances that a given Booster Pack board may be used with more than one type of Launch Pad board (see the Pinout Standard document for information).

Figure 3 of the User's Guide indicates how GPIO16-17 and GPIO32-33 are connected to headers J2 and J6. First of all, note that the schematic diagram shows 8 jumpers, labeled JP4-JP11. These jumpers are not traditional jumpers that can be easily installed or removed by hand; instead, they are actually hard-wired connections that have been made on the board's top copper layer. The image below displays a zoomed view of the board's electrical connections in the region of interest; the top copper connections are shown in red, the bottom copper connections are shown in blue, the vias connecting the two copper layers are shown in yellow, and the pads for headers J2 and J6 are shown in green. The non-conventional jumpers appear in this image as 8 red vertical connections between 8 pairs of red square pads. Any of these non-conventional jumpers may be broken with a knife, and then reconnected later if needed using 0 Ω resistors (0402 size).



From the schematic diagram, it is also clear that GPIO16 and GPIO32 have been shorted together and that GPIO17 and GPIO33 have been shorted together; moreover, the GPIO16/32 shorted node is connected to two header pins (J2-6 and J6-7) and the GPIO17/33 shorted node is connected to two header pins (J2-7 and J6-8). This arrangement appears at first glance to be both redundant and limiting. To make sense of this design, recall that all GPIO pins are automatically initialized as high impedance inputs during the boot process. For this reason, it is possible to use GPIO16-17 for SPI functionality, or to use GPIO32-33 for I2C functionality, without breaking any jumpers; no contention would occur since two of the four pins involved would remain as high-impedance inputs.[3] If an application requires the simultaneous use of SPI and I2C communications, then it would be necessary to break some jumpers such that the SPI pins will be connected to one header and the I2C pins will be connected to the other header.

[2] See `http://www.ti.com/ww/en/launchpad/launchpad.html?DCMP=mcu-launchpad&HQS=launchpad`.

[3] SPI and I2C are serial communication standards commonly used to provide chip-to-chip data exchange.