Slice Testboard v1.1 USB Interface and E-Fusing Instructions

2021-09-23

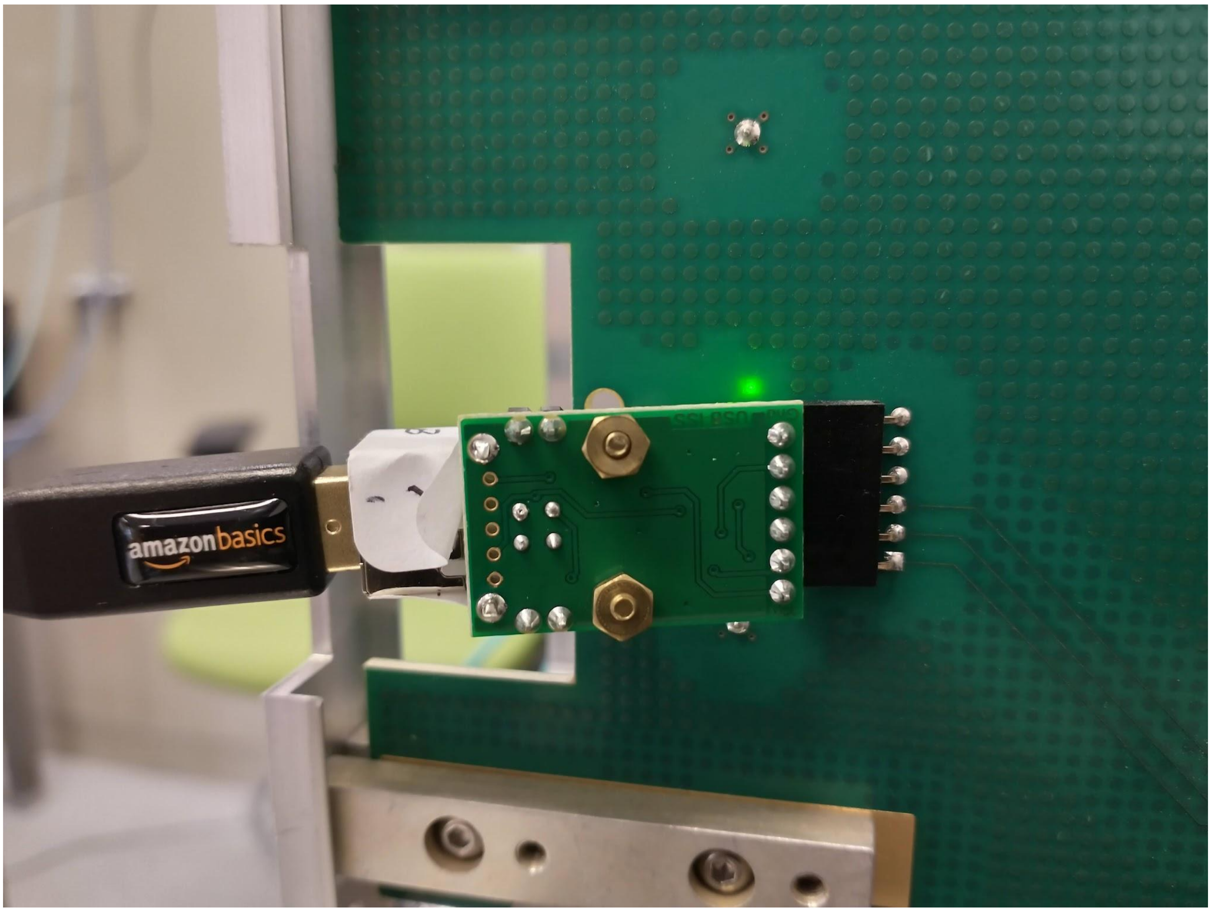
This document provides instructions to set up the USB interface to a v1.1 slice testboard, to configure lpGBT11,12,13,14 through this interface and to perform the lpGBT e-fusing process.

Prepare USB dongle

-place header on left-side of USB connector to enable 5V operation
-see USB dongle documentation on twiki for more details

Connect USB dongle to Slice Testboard v1.1 USB connector

-dongle attaches to header with mounting screws pointed towards slice testboard surface



Software Setup

-checkout out slice-testboard project

-switch to usb_I2C_GUI branch

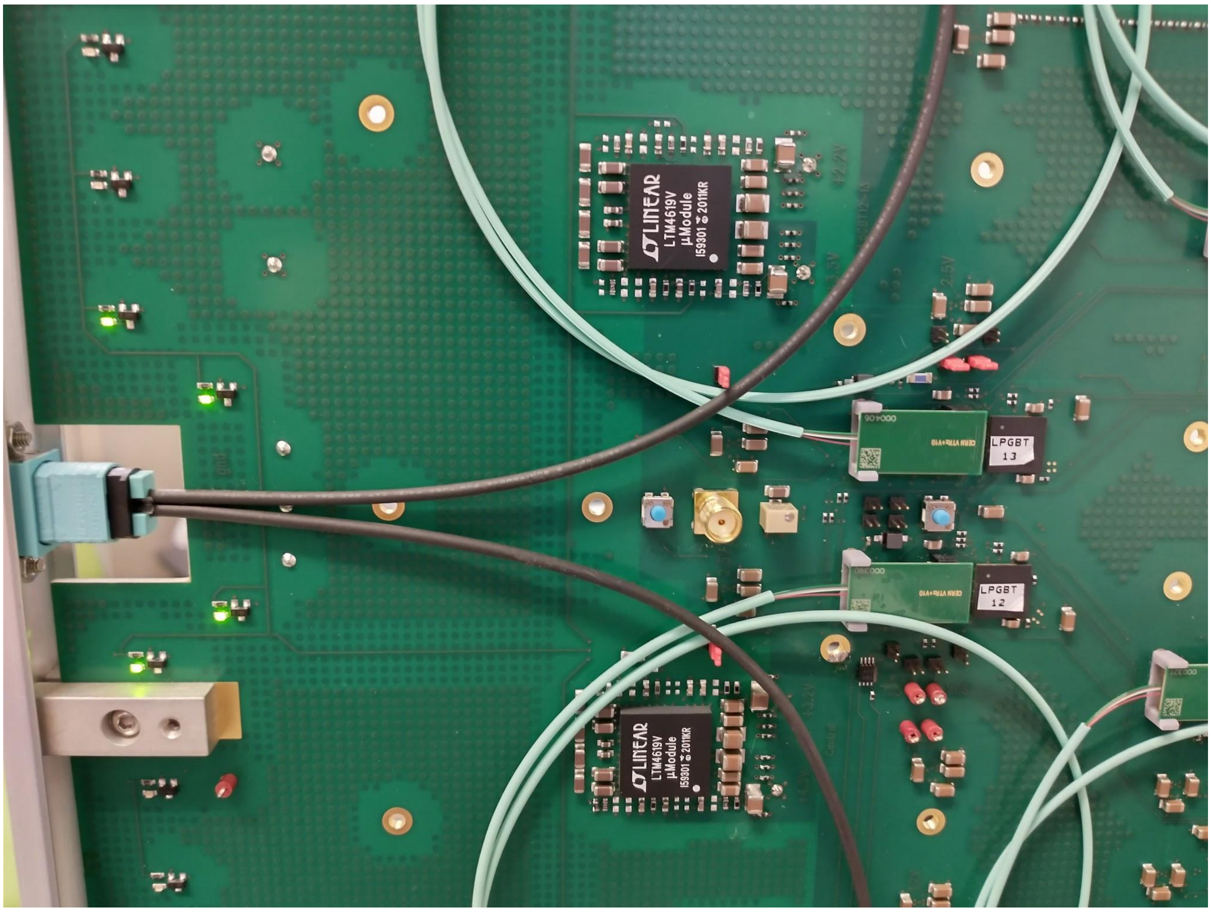
Header Configuration Summary Table

Header Name	Default IC, no USB	lpGBT12 on USB	lpGBT13 on USB
H15	OFF	ON	OFF
H16	OFF	ON	OFF
H20	ON	OFF	OFF
H21	ON	OFF	OFF
H22	ON	OFF	OFF
H23	ON	OFF	OFF
H24	OFF	DC	ON
H25	OFF	ON	DC
H40	OFF	OFF	ON
H41	OFF	OFF	ON

-see header setting instructions on twiki for more details

Connect USB interface to control LPGBT12

-means connecting USB SDA/SCL to lpGBT13 M2 SDA/SCL bus
-see table for header placement, set headers for “lpGBT12 on USB”, should look like:



-run basic USB interface script:

```
python configureLpGBT1213.py 12
```

-on a Linux system if there’s a permission error try giving the interface permissions via:

```
chmod 666 <interface name>
```

-script should report if the lpGBT was detected on the I2C bus

Configure lpGBT12 registers with “minimal” config via USB

-run basic interface script:

```
python configureLpGBT1213.py 12 -c
```

-should see a sequence of USB operations to do lpGBT12 register writes and readbacks

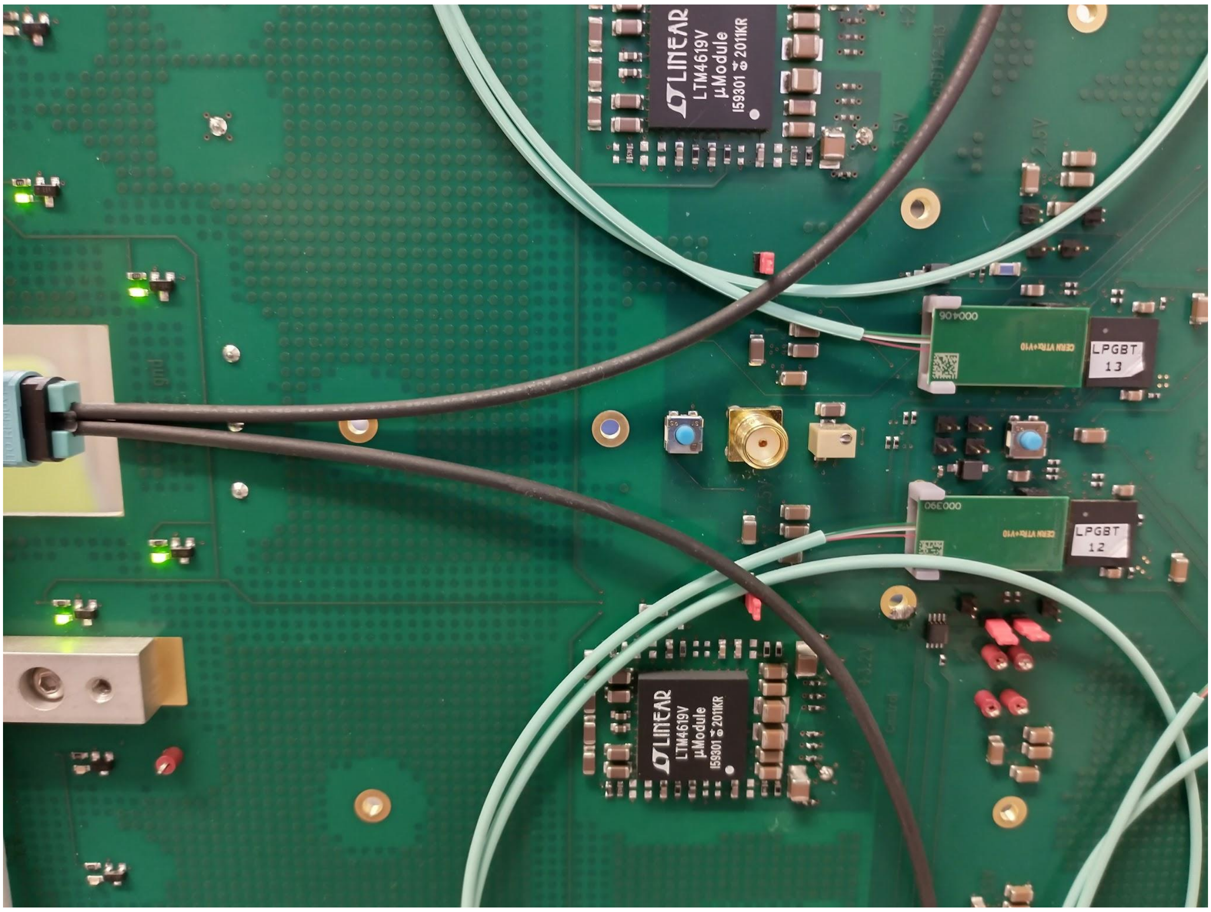
-the script should report if the register value was read back correctly

-if all readbacks successful then lpGBT12 configured, can run e-fusing process

Connect USB interface to control lpGBT13

-means connecting USB SDA/SCL to lpGBT12 M2 SDA/SCL bus

-similar to lpGBT12 case, except now headers are set to “lpGBT13 on USB” configuration:



-run basic USB interface program:

```
python configureLpGBT1213.py 13
```

Configure lpGBT13 registers with “minimal” config via USB

-run basic interface script:

```
python configureLpGBT1213.py 13 -c
```

-should see if writes are successful via readbacks

Configure lpGBT11 and lpGBT14

-extra steps required to enable lpGBT11 and lpGBT14 I2C
-lpGBT12 and lpGBT13 must be configured as described above
-lpGBT12 and lpGBT13 must be receiving a 640MHz reference clock via the FELIX downlink and ready LEDs should be ON

-connect to lpGBT12 USB interface as above
-run the lpGBT12 and lpGBT13 I2C enable script:

```
python enableLpGBT1114i2c.py 12
```

-connect to lpGBT13 USB interface as above
-run the lpGBT12 and lpGBT13 I2C enable script:

```
python enableLpGBT1114i2c.py 13
```

-should now be able to configure lpGBT14 via USB I2C:

```
python configureLpGBT1213.py 14 -c
```

-to configure lpGBT11, connect to LPGBT12 USB interface again:

```
python configureLpGBT1213.py 11 -c
```

E-Fusing lpGBT registers

-connect USB interface and run configure script as described already, but include the -f option and follow directions on terminal

-for example to configure lpGBT12 connect USB interface to lpGBT12 and run:

```
python configureLpGBT1213.py 12 -c -f
```

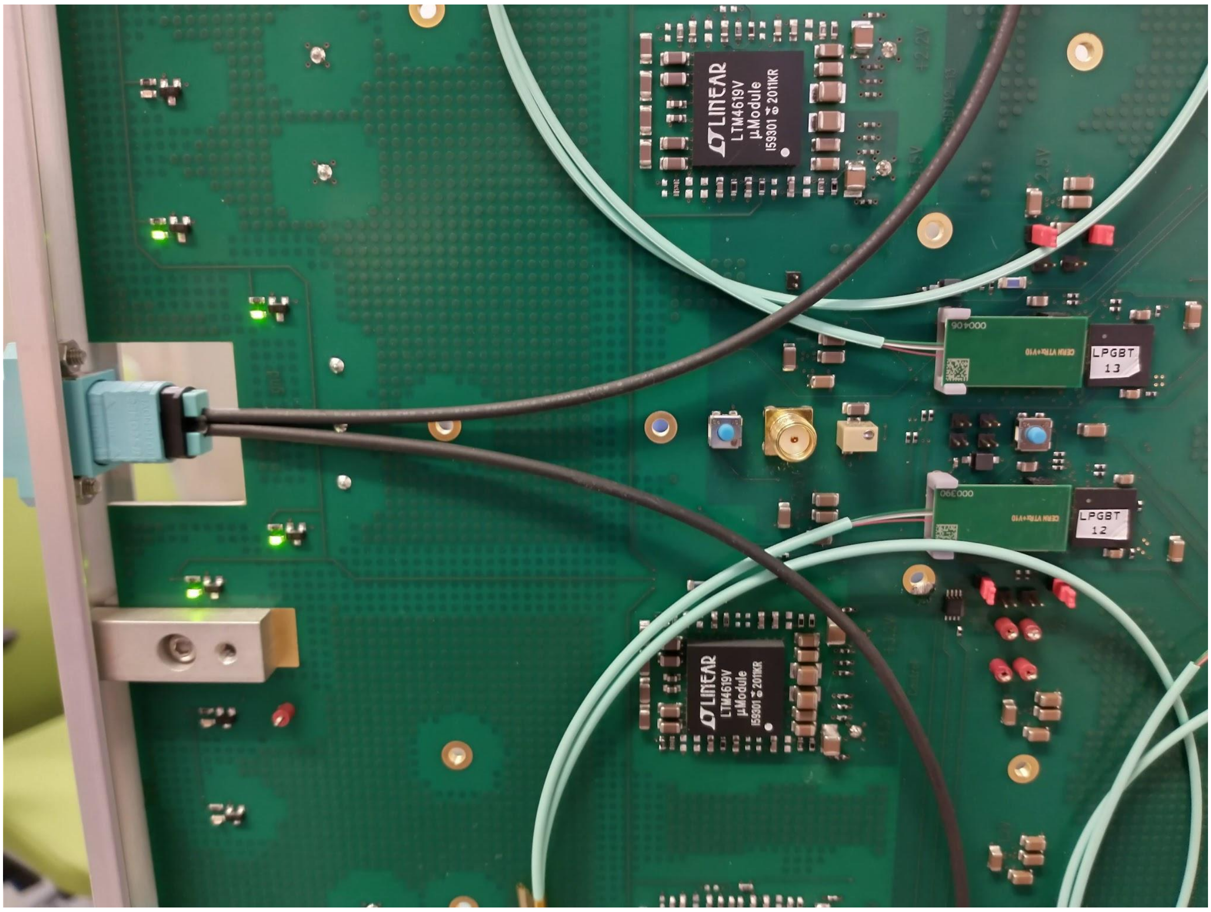
-when the script writes to terminal “"Press enter once VDDF2V5 is on” press and hold SW2, do not let go until the script writes “Press enter once VDDF2V5 if off”

-attach picture

Returning to IC Interface

-remove USB dongle from slice testboard

-set headers to “Default IC, no USB” as shown in table:



-if lpGBTs were successfully e-fused, then on power-cycling the lpGBT12,13 ready LEDs should turn on if the FELIX downlink clock is connected