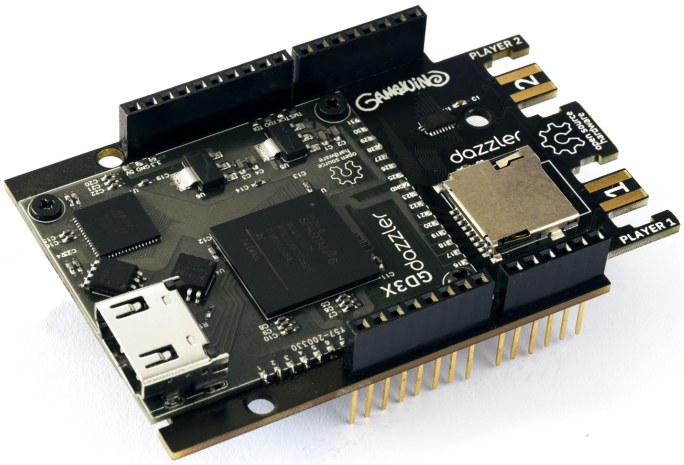
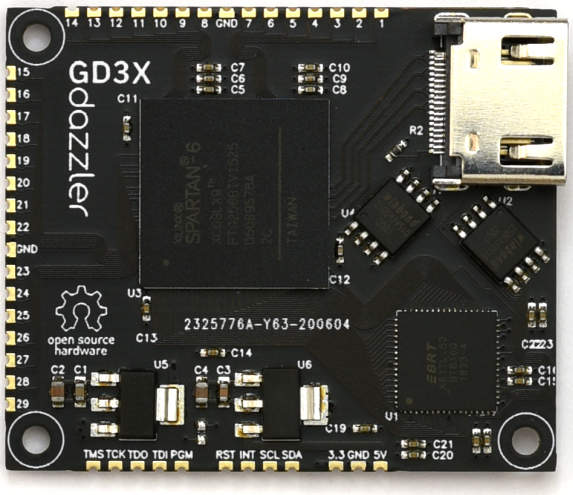


Last updated on January 10, 2021

Contents

1 Overview



The Gameduino 3X Dazzler outputs HD picture and sound to any HDMI display. It is available as a core module, and as an Arduino-compatible shield.

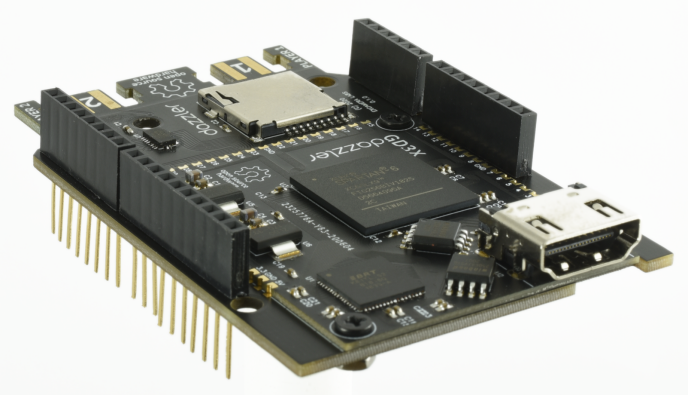
1.1 Features

The core module has the following features:

- Powerful BT815 embedded GPU with 24 bit color
- HDMI encoding and system management handled by Xilinx Spartan-6 LX9 FPGA
- 2 $\times$ 8 Megabyte flash
- HDMI video output at 1280x720 @ 60 Hz (720p), audio output at 48 KHz
- All features accessible via SPI interface
- Single 5V supply; onboard regulation to 3.3V and 1.2V

The shield adds:

- a level shifter so all inputs are 5V tolerant
- a microSD card slot
- 2 Wii Classic ports, both continuously scanned by the Dazzler
- optional headers for standard serial and JTAG connectors



2 Getting Started

On power-up, the Dazzler initializes the BT815 and starts HDMI output, and enables the SPI interface. Note that many monitors take several seconds to detect and sync to an HDMI signal.

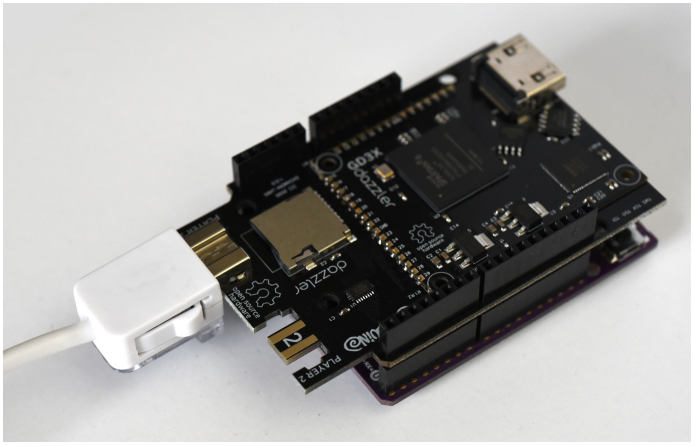
If no SPI commands are sent by the MCU, the Dazzler displays a boot screen like this:

The code on the left is the module's serial number. The center code is the firmware version. On the right is the firmware build date.

The Dazzler functions like the other Gameduino 3X family members. With an Arduino or Teensy, the Gameduino library supports all Gameduino 3X series devices:

	resolution	touch input	Wii-classic input
Gameduino 3X 4.3"	480×272	•	
Gameduino 3X 7"	800×480	•	
Gameduino 3X Dazzler	1280×720		•

Connect Wii-classic compatible game controllers as shown.



The standard Gameduino library is available at:

https://gameduino.com/code

The CircuitPython library also supports the Gameduino 3X devices. It is available at:

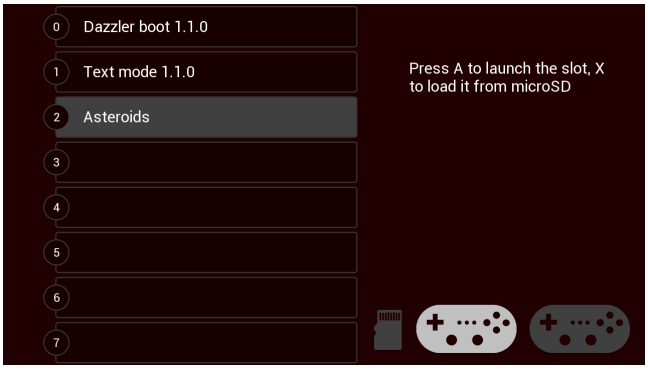
https://github.com/jamesbowman/py-bteve

2.1 Accessing the boot menu

If you have a Wii-classic compatible controller, you can access the Dazzler's built-in boot menu at any time.

Attach the controller to the "PLAYER 1" port, and press **START +**, **x**, **y** buttons at the same time.

The display will switch to the boot menu.



The three icons on the right indicate presence on the microSD, PLAYER 1 and PLAYER 2 slots respectively.

Use the left-direction pad to select one of the eight available flash slots. Button **a** launches a slot.

Dazzler organizes the 8 Mbyte FPGA boot flash into 8 **slots**. Each slot can contain a boot image. Slot 0 is the boot slot, and should contain a Dazzler boot

image. Other slots can be loaded with alternate firmware images. To install an image in a slot:

1. use a microSD card formatted with FAT32
2. copy the flash image to the microSD card naming it `image.dazzler`
3. insert the card into the Dazzler's microSD slot, and connect the controller
4. activate the boot menu
5. select a blank slot, then press **x** to start the write process
6. the write process takes a few seconds. After it completes the boot menu shows the new slot contents

3 SPI interfaces

3.1 Gameduino 3X and microSD

Like other Gameduino 3X family members, Dazzler has BT815 and microSD SPI interfaces. See section 6 Pinouts for details. These function the same way as all Gameduino SPI pins, with a direct connection to the BT815 and to a microSD card in SPI mode.

3.2 Dazzler SPI interface

The Dazzler has its own SPI interface distinct from the BT815 and microSD SPI interfaces. This interface controls the Dazzler at a system level and allows firmware updates.

SPI writes to the Dazzler use the following format. Additional bytes are ignored.

+0	command
+1	N

SPI reads from the Dazzler use the following format.

+0	ID byte: `0xDA`
+1	status byte
+2 .. +7	Player 1 joystick
..	
+14 .. +19	Player 2 joystick

The status byte is `0xff` while the Dazzler is busy processing a command. The status byte is `0x00` when the Dazzler is ready to accept a command.

The following commands are supported:

`0x40` Reset the BT815

`0x41` Boot from slot **N**

`0x42` Load slot **N** from file `image.dazzler` on microSD

All other byte values are ignored.

The two 6-byte joystick fields follow the Wii Classic controller bit definitions:

+0	RX[4:3]			LX[5:0]				
+1	RX[2:1]			LY[5:0]				
+2	RX[0]	LT[4:3]		RY[4:0]				
+3	LT[2:0]			RT[4:0]				
+4	BDR	BDD	BLT	B-	BH	B+	BRT	1
+5	BZL	BB	BY	BA	BX	BZR	BDL	BDU

LX,LY are the left Analog Stick X and Y (0-63), RX and RY are the right Analog Stick X and Y (0-31), and LT and RT are the Left and Right Buttons (0 or 31). The left Analog Stick has twice the precision of the other analog values.

BDL,R,U,D are the D-Pad direction buttons. BZR,ZL,A,B,X,Y,+,H,- are the discrete buttons. BLT,RT are the digital button click of LT and RT. All buttons are 0 when pressed.

The above table and text are taken from

https://wiibrew.org/wiki/Wiimote/Extension_Controllers/Classic_Controller_Pro.

4 Text mode

In text mode the Dazzler's onboard hardware directly drives the BT815 GPU to produce a high-quality text display on the HDMI output. The input is a single serial line running at 115200-8-N-1.

The terminal emulation supports standard ANSI escape codes, including 256-color support. Multiple screen modes are available:

0	128×34	landscape
1	182×48	landscape
2	80×64	portrait
3	102×85	portrait

On entry, text mode displays a blank screen with a blinking cursor. Any data received on the serial line is immediately drawn on the screen.

4.1 Entering text mode

Text mode is installed in slot 1 of the Dazzler's boot flash. It can be selected by:

- selecting slot 1 from the boot menu. See subsection 2.1 Accessing the boot menu
- The Dazzler serial console. Use command `1 run`. See section 5 Console
- The Dazzler SPI interface. See subsection 3.2 Dazzler SPI interface and the sample code below.

On Arduino and compatible microcontrollers, an SPI command to boot the Dazzler from slot 1 is:

```
#include <SPI.h>

void setup()
{
  Serial.begin(115200);
  SPI.begin();
  for (byte i = 8; i <= 10; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, HIGH);
  }
  digitalWrite(10, LOW);
  SPI.transfer(0x41);     // Boot from slot
  SPI.transfer(0x01);     // slot 1
  digitalWrite(10, HIGH);
}

void loop()
{
  Serial.println("Hello world");
  delay(500);
}
```

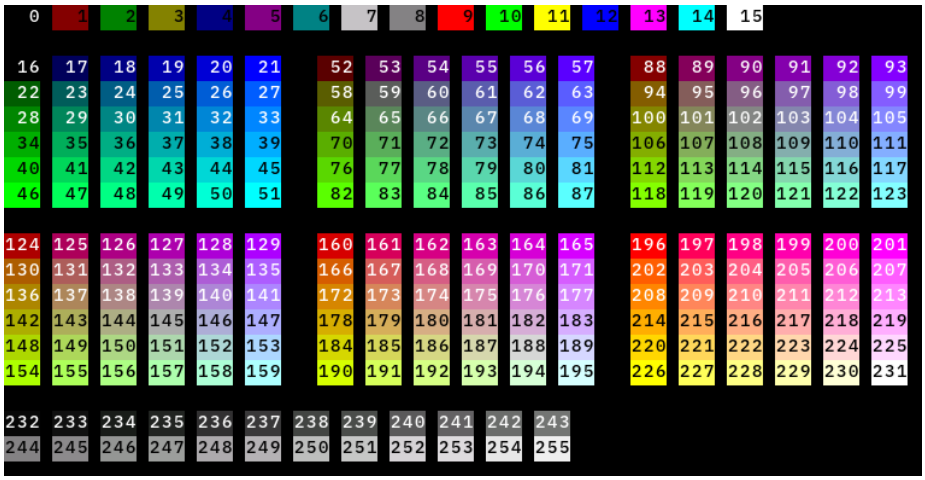
On entering text mode, the default layout is 128x34 landscape. The display is blank except for a blinking cursor.

4.2 ANSI escape codes

The following standard CSI codes are supported:

Code	Effect
ESC [*n* A	Cursor up
ESC [*n* B	Cursor down
ESC [*n* C	Cursor forward
ESC [*n* D	Cursor back
ESC [*r;c* H	Cursor position
ESC [*n* J	Erase display
ESC [*n* m	Select graphic rendition (SGR)
ESC [s	Save cursor position
ESC [u	Restore cursor position

The SGR command supports ANSI standard 256 color mode.

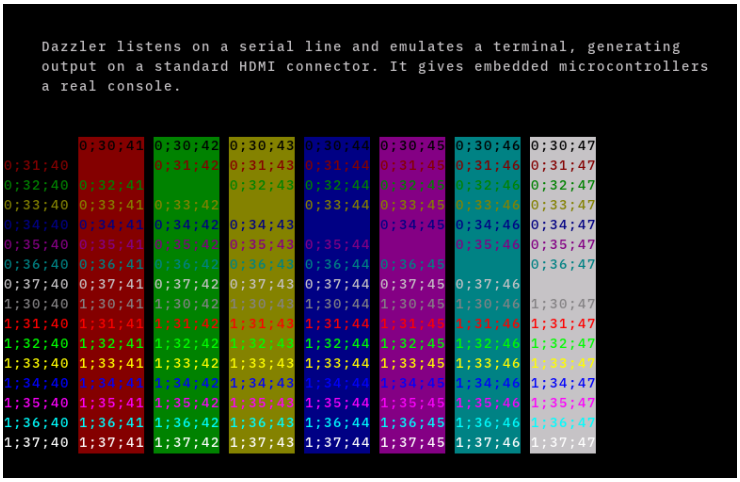


In addition the following sequences are specific to Dazzler's text mode:

Code	Effect
ESC [*n* h	Set display mode. 0 is 128x34 (landscape), 1 is 182x48 (landscape), 2 is 80x64 (portrait), 3 is 102x85 (portrait)
ESC [*n* S	Screen-saver. 0 stops video output, 1 restarts video output

For example this C program displays the 16 ANSI foreground and background colors.

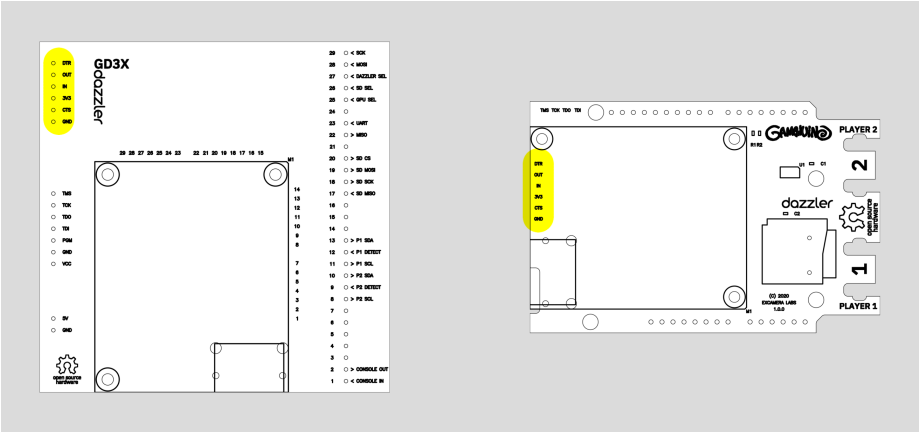
```c
#include <stdio.h>
int main()
{
  int attr, fg, bg;
  printf( "Dazzler listens on the serial line and emulates a "
   "terminal, generating\noutput on a standard HDMI connector."
   "It gives embedded microcontrollers\na real console.\n\n");
  for (attr = 0; attr <= 1; attr++)
    for (fg = 30; fg <= 37; fg++) {
      for (bg = 40; bg <= 47; bg++)
        printf("\e[%d;%d;%dm%d;%d;%d\e[m",
          attr, bg, fg, attr, bg, fg);
      printf("\n");
    }
  return 0;
}
```



5 Console

The serial console uses pins 1 and 2 on the Dazzler module.

Both the Arduino Shield and the Module Breakout have connectors for 6-pin FTDI USB connectors. All signals are 3.3V. Connecting to the Arduino shield requires a 6-pin header soldered to the bottom of the shield. See subsection 6.1 Shield and subsection 6.3 Module Breakout.



The console is always running, and the connection speed is 1000000 8-N-1.

To access it, connect to the port and type ctrl-C. Some useful console commands are listed below. Command parameters always precede the command.

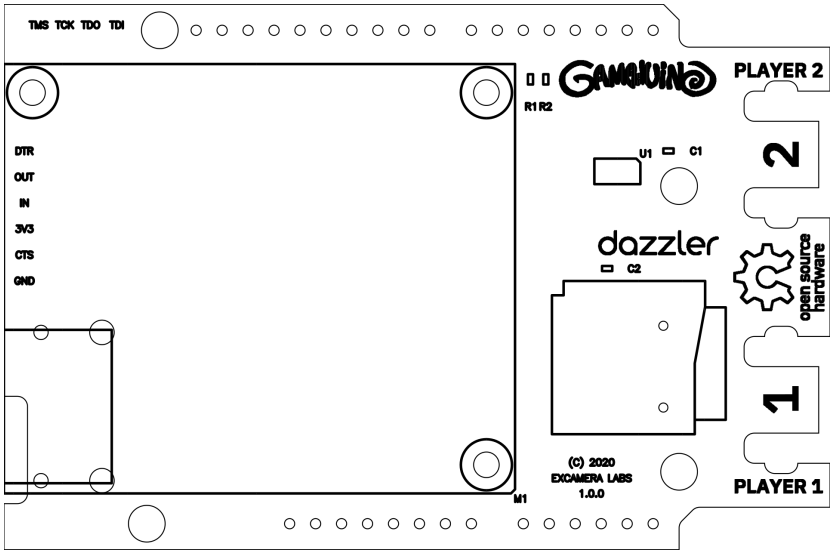
`slots`	list the current slot contents
`reboot`	boot from slot 0
s `run`	boot from slot s
s `zslot`	erase slot s
s `loadsd`	load slot s from microSD file `image.dazzler`

The system runs SwapForth on a 50 MHz 16-bit J1 soft-core CPU.

6 Pinouts

6.1 Shield

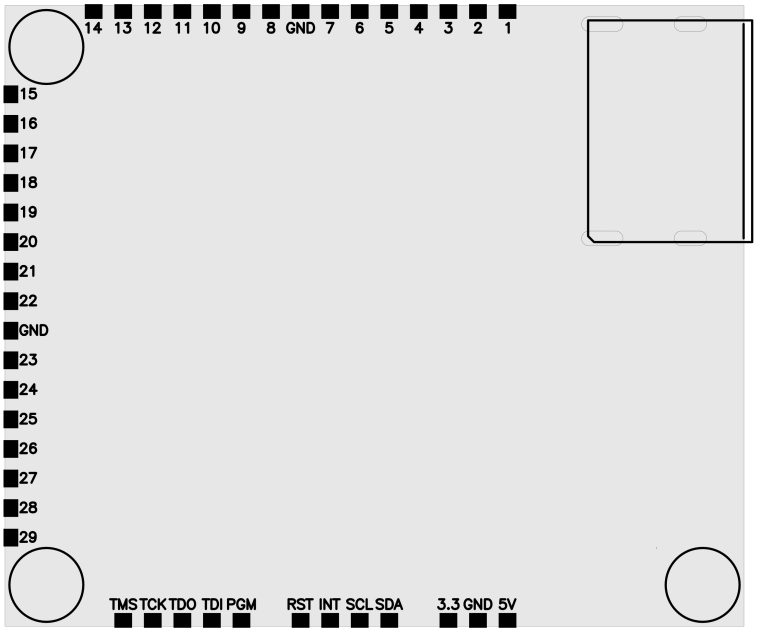
The Dazzler Shield follows the standard Arduino Uno pinout.



GND	power	Signal ground
5V	power	Main supply: 5-9V
13	in	SPI SCK
12	out	SPI MISO
11	in	SPI MOSI
10	in	DAZZLER SEL
9	in	SD SEL
8	in	GPU SEL
2	out	INTERRUPT
1	in	SERIAL IN

All other pins are pass-through.

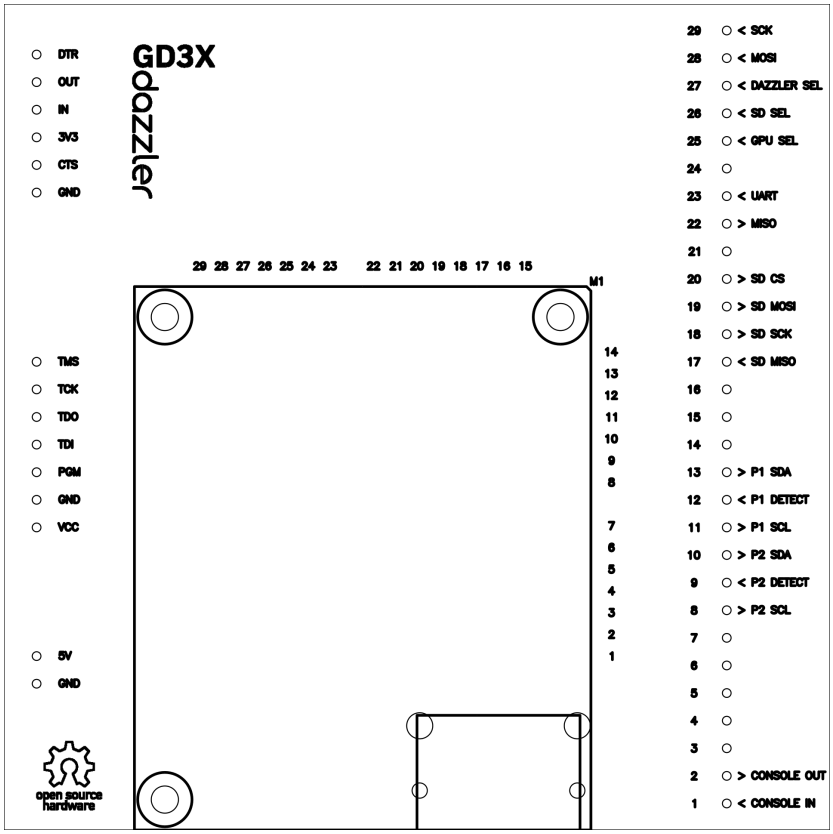
6.2 Module



pin	group	direction	function
GND	power	in	Signal ground
5V	power	in	Main supply: 5-9 V
3.3	power	out	3.3 V output
1	console	in	CONSOLE IN
2	console	out	CONSOLE OUT

pin	group	direction	function
8	Wii	in/out	P2 SCL
9	Wii	in	P2 DETECT
10	Wii	in/out	P2 SDA
11	Wii	in/out	P1 SCL
12	Wii	in	P1 DETECT
13	Wii	in/out	P1 SDA
17	microSD	in	SD MISO
18	microSD	out	SD SCK
19	microSD	out	SD MOSI
20	microSD	out	SD CS
22	SPI	out	MISO
25	SPI	in	$\overline{\text{GPU SEL}}$
26	SPI	in	$\overline{\text{SD SEL}}$
27	SPI	in	$\overline{\text{DAZZLER SEL}}$
28	SPI	in	MOSI
29	SPI	in	SCK

6.3 Module Breakout



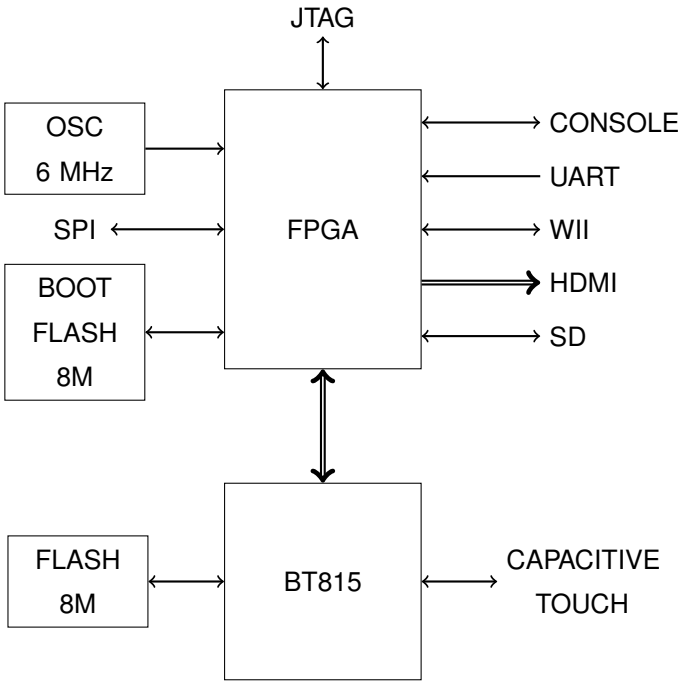
The module breakout board has the same pin numbering as the module, but uses 0.1” headers for convenience.

The left-hand side has:

- a 6-pin header for standard FTDI USB-UART boards
- a 7-pin JTAG header
- 5V power header

Note that 3.3V (VCC) is available on the JTAG header.

7 Internals



The FPGA used in the Dazzler is a Xilinx Spartan 6 XC6SLX9-FTG256. It has a high-bandwidth connection to the GPU, a BridgeTek BT815.

Both FPGA and GPU have 8 Mbyte boot flashes.

The FPGA UCF file for the board, and all Verilog, firmware and build files are provided at:

https://github.com/jamesbowman/gd3x-dazzler

8 Specifications

8.1 DC characteristics

	min	typ	max	units
All input signals				
low voltage	-0.5		0.8	V
high voltage	2.0		4.1	V
Supply voltage	4.5	5.0	9.0	V
Current consumption		180		mA

8.2 AC characteristics

	min	typ	max	units
SPI speed			36	Mbps
Startup time			270	ms
Scanout frequency		74.25 $\pm$ 0.002%		MHz
Scanout frame rate		60.000		Hz

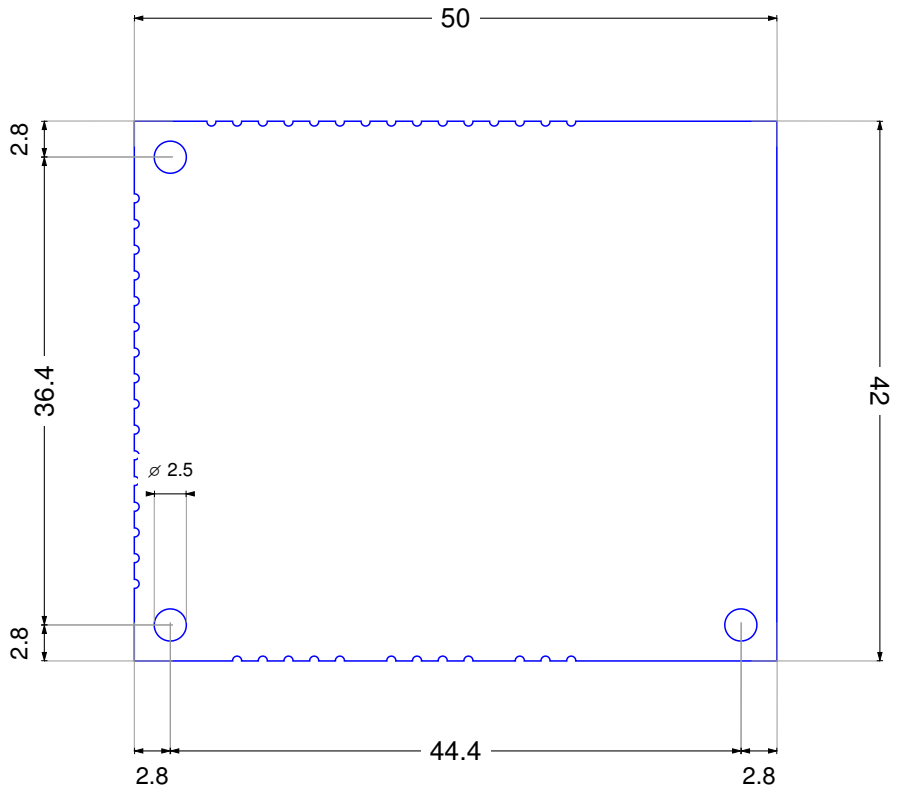
8.3 Physical characteristics

		units
Dimensions (shield)	83 × 53 × 20	mm
Weight (shield)	32	g
Dimensions (module)	50 × 42 × 8.5	mm
Weight (module)	10	g

8.4 Module mechanical drawing

All measurements are in mm.

The three mounting holes are designed for M2.5 screws, nuts and standoffs.



9 Troubleshooting

No monitor output after power-up	• Check HDMI cable and port • Confirm 5V power present • Contact support
No audio output	• Confirm the display has audio output • Check the display's mute and volume settings
Corrupted video signal	• Use a 2 m (6 foot) HDMI cable
Wii controller unresponsive	• Check the plug orientation

10 Support information

Technical and product support is available at jamesb@excamera.com

Gameduino 3X Dazzler is built and supported by Excamera Labs.

Index