GameKid

An Open-Source Handheld Gaming Device
Trent Dye, Spring 2018
https://github.com/trentdye/gamekid

The GameKid is a portable gaming device inspired by video game systems like the original Nintendo Game Boy. I designed and built the GameKid to explore the topic of integrated product design for an Olin Self-Study engineering elective. The project helped refine my skills in circuit design, PCB/PCA design, design for 3D printing, and Arduino/C++.

Background

I decided to design and prototype a handheld gaming device because I knew it would be a fun yet challenging project. I particularly liked the idea of designing a gaming device because there are a few basic requirements that most people would agree are necessary features of a portable Gameboy-like system. I knew my device would have to have:

- an easy-to-see display,
- buttons that are easy to press,
- an on/off switch,
- a form factor that would make it fit well in the hand and isn't unreasonably heavy,
- the necessary processing power to play a decently challenging game,
- and a replaceable or rechargeable battery.

Having these requirements in mind allowed me to constrain the project enough to allow me to finish it in time. In order to keep this project properly scoped for a semester of work, I added only a few more features to these "basic" but challenging feature requirements: USB charging and indicator lights.

Once the features had been established, the design of the product began with choosing two of the components that most dictated the design of the product: the microcontroller and the LCD screen. My choices were decided by creating and testing prototypes until I liked what I had. Following these decisions, I created a schematic encompassing my entire system. During this process, I selected the ICs and components I needed to build my entire circuit. After iterating on the design of the electrical design until I was happy with it, I selected the electromechanical components – the buttons and switches. From there, I began to design the PCB. Because I was designing a PCB for the first time, the design process took longer than expected, requiring a significant number of design/review cycles with my adviser. After sending out the PCB to be manufactured, I set to work on the design of the enclosure. For the rest of the semester, I transitioned between working on mechanical design, electrical assembly, and firmware development in order to integrate everything into a playable device.

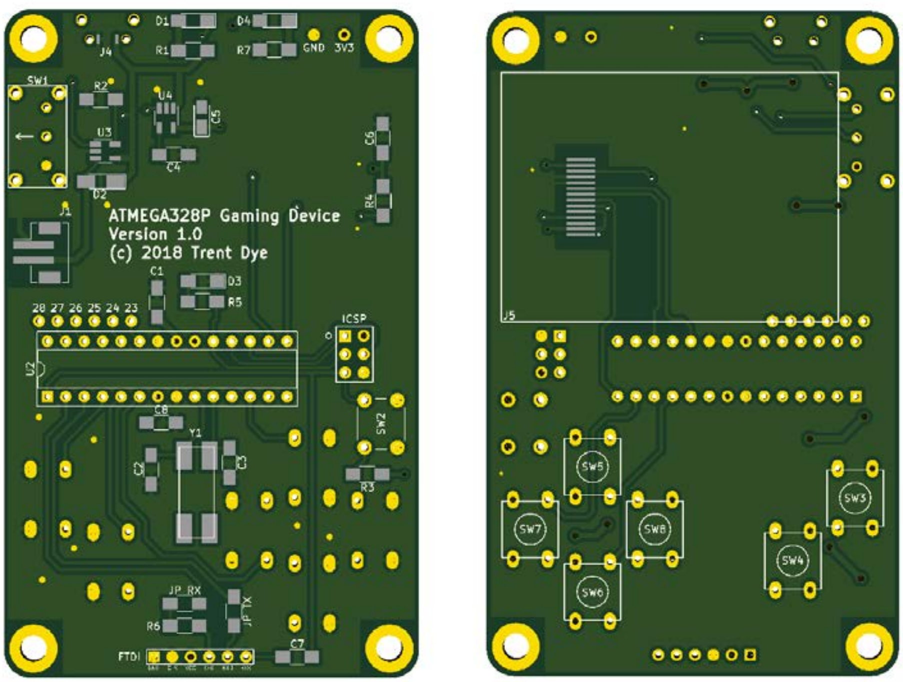


FIGURE 1: RENDERS OF THE UNPOPULATED PCB, FROM KICAD.

ELECTRICAL

The attached schematic details the entirety of the electrical system, which was routed on a single PCB.

Design Choices

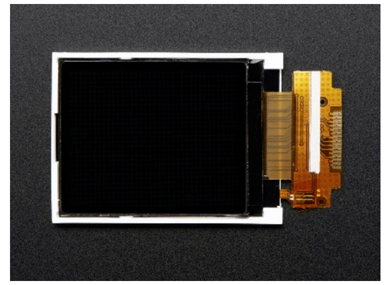


FIGURE 2: JDT-1600 LCD. IMAGE COURTESY: ADAFRUIT.COM

- **LCD Screen**: I decided on the JDT-1600 because of its availability in a ribbon package in single quantities, shipped from the US via Adafruit. Although it was smaller than I was initially hoping, I was able to adapt the design of the board to fit the form factor of the screen. As an added bonus, the ST7735R driver included with the board allowed it to be run with minimal wiring and memory usage and compatible with an Adafruit-written display library.
- **Microcontroller**: I chose the ATMega328P for a variety of reasons. Familiar with the microcontroller from Arduino devices, I was able to figure out how to program it fairly easily using the Arduino environment. For a first-pass board, it was imperative that I choose a microcontroller in a DIP package so that I could use a breadboard to flash a bootloader to the chip before plugging it into the PCB before I knew whether my ICSP header could reliably be used to do the task. Having limited experience in choosing and using microcontrollers I picked an implementation I was familiar with: running the ATMega328P at 8Mhz and 3.3V as the 3.3V Arduino Pro Mini does.
- **Battery Charging Circuit**: To find a suitable battery charging circuit for my device, I looked at the schematics of similar devices. Borrowing the design of Sparkfun's Lilypad was ideal, since it was achieving my goal: running an ATMEGA328P and peripherals at 3.3V. The design uses a MCP73831 in conjunction with a diode and a 3.3V LDO. This circuit allowed the device to perform as one would expect: power could be provided either from the battery or directly from the USB port and the battery could be charged either while the device was powered off or while still playable.
- **Battery**: Unsure of the amount of power that my device would consume, I purchased a 1200mAh LiPo battery to use to test the first iteration of the device. After testing, I concluded that the GameKid uses 21mA during normal operation and less than 10uA when powered off. To achieve 5 hours of on-state battery life with an worst-case expected battery utilization of 60% (given that I use an LDO and no boost converting circuitry used to produce a 3.3V rail), I needed a 175mAh or greater battery. I settled on a 350mAh battery after learning that a smaller battery than that would not allow me to reduce the thickness of the enclosure.

Electrical Schematic

Please see the attached electrical schematic for a diagram of the entire electrical system in the gaming device. The schematic was designed using KiCAD Eeschema schematic creator. A KiCAD project file is available in the Github repository for download.

PCB

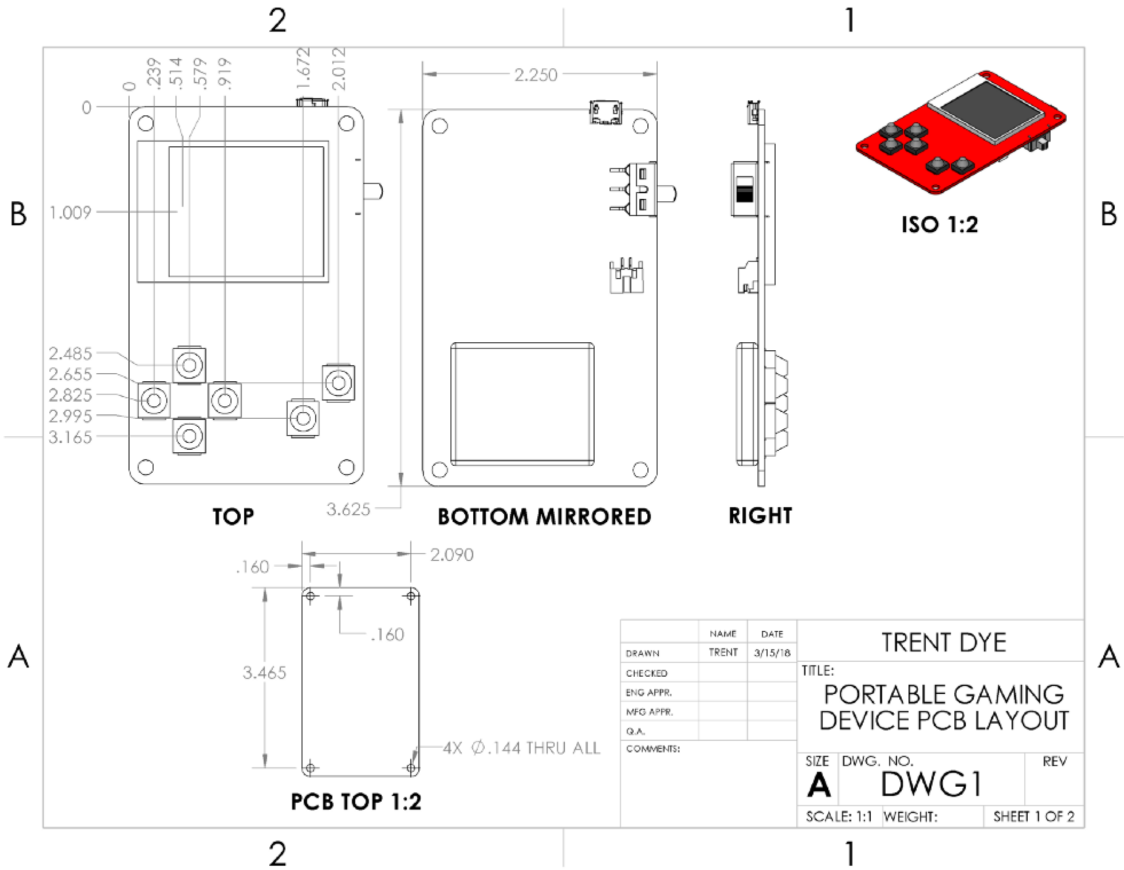


FIGURE 3: DESIGN OF A BASIC LAYOUT OF THE PRINTED CIRCUIT ASSEMBLY, USED FOR THE PLACEMENT OF PCB FOOTPRINTS.

The PCB design process began with modelling the I/O components—screen, buttons, USB port, status lights—in CAD and arranging everything on a rectangle of reasonable size for the PCB (Figure 3). With the placement of those components established, the rest of the components were placed in orientations that would most simplify the PCB routing. Based on the generous predicted size constraints of the device, the entire electrical system could fit on a single two-layer PCB. PCBs were ordered through PCBWay. Gerber files, as well as a KiCAD project file, are available in the Github repository. A rendering of the PCB is shown in Figure 1.

Bill of Materials

Please see the attached bill of materials for a list of parts that need to be purchased in order to assemble a GameKid.

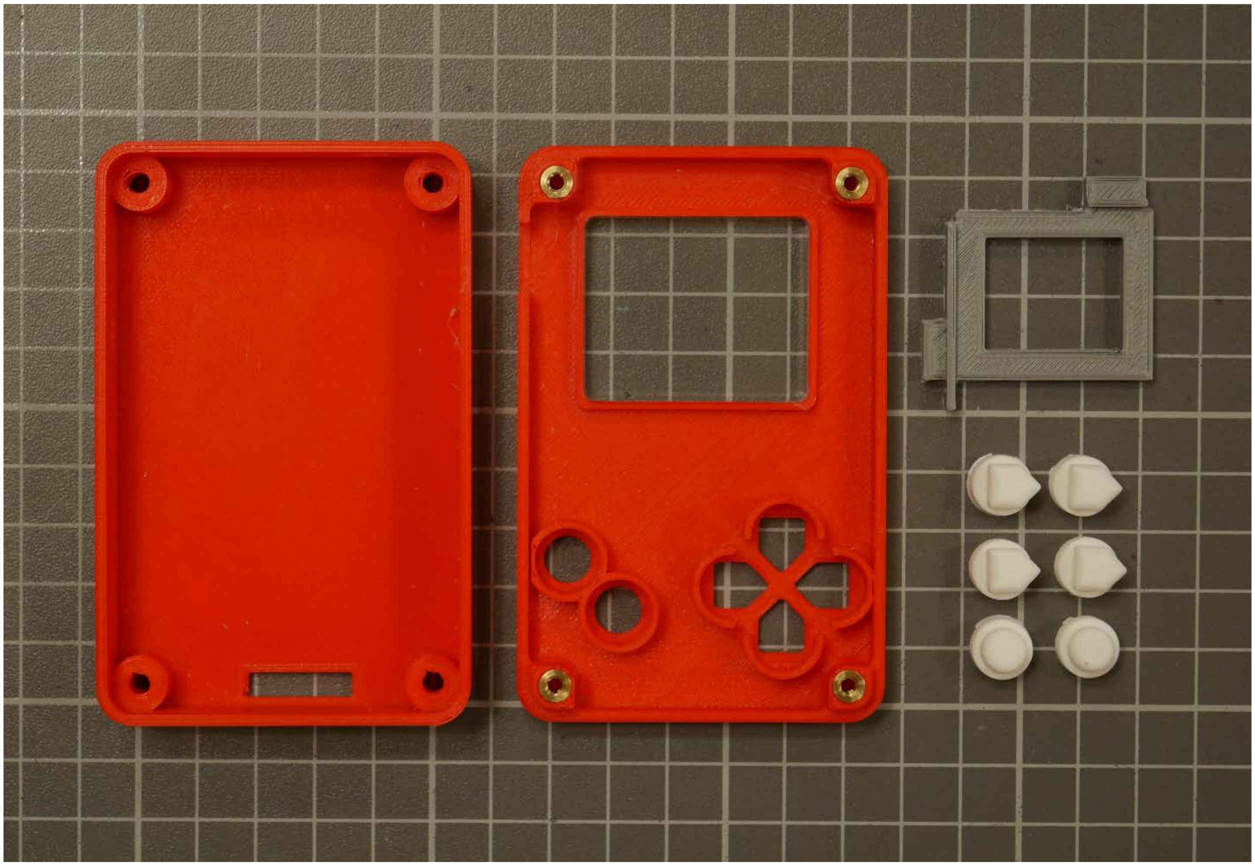


FIGURE 4: THE 3D PRINTED COMPONENTS OF THE MODULE.

MECHANICAL DESIGN AND ASSEMBLY

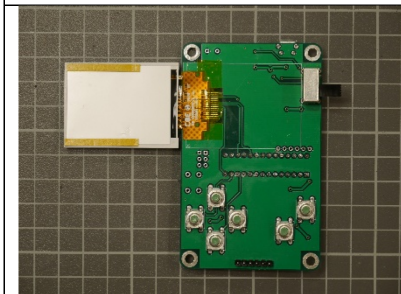
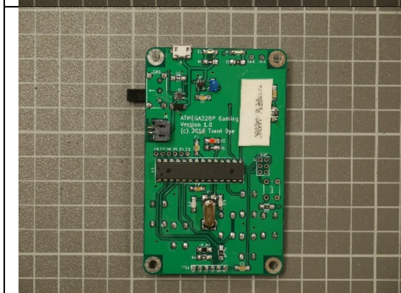
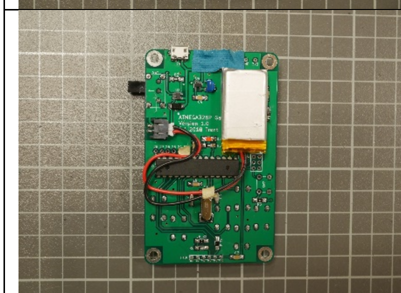
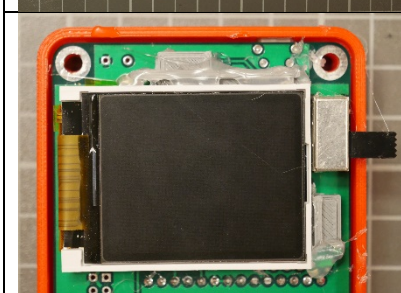
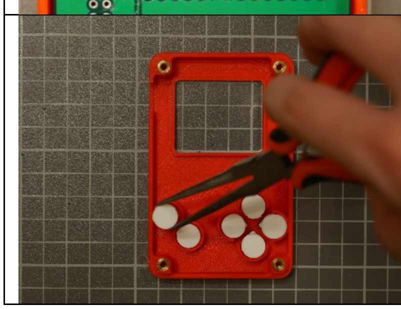
The enclosure was designed to be entirely 3D printed, with the exception of the clear screen protector, which is optional and designed to be laser cut out of clear acrylic. The 3D printed components are shown in Figure 3. For the 3D printed components, STL files are available in the Github repository, and a DXF file is available for the laser cut component.

FABRICATION INSTRUCTIONS

Use a 3D printer to print “Case Bottom,” “Case Top,” “Button Directional,” and “Button Nondirectional” in the quantities specified on the attached bill of materials. My parts were printed on a Prusa i3MkII using an 8mm Brim printing support and 30% infill.

Use a laser cutter to cut “Screen Protector” out of 1/8” clear acrylic using the recommended settings for your machine.

Assembly Instructions

	1. Assemble PCB, omitting the optional SW2 and J3 components.
	2. Use Kapton tape to secure ribbon cable.
	3. Place foam tape for battery mounting as shown.
	4. Install battery and route wire as shown. Place gaffer’s tape to obscure LED light. Ensure that light can still come through the top of the package.
	5. Install PCB into case bottom. Place carrier on board, and align LCD screen to carrier. Using hot glue, attach carrier to PCB and PCB to screen such that screen is aligned to silkscreen outline on PCB.
	6. Press standoffs and screen into case top. Place buttons in respective locations.

	7. Install the two case halves together as shown, such that buttons do not fall out of case.
	8. Install screws and tighten with 2.5mm allen key.
	9. Verify that buttons can fully actuate, case bottom and top come together, and screen is oriented with enclosure. Make adjustments if necessary.

SOFTWARE AND PROGRAMMING

The choice of the ATMega328P was made for compatibility with the Arduino development environment, which allowed for quick development using several premade libraries. However, the device may be programmed in C using Atmel Studio. The device contains an FTDI (serial UART) header as well as an ICSP (SPI) header for programming with an FTDI programmer or an AVRISP or similar device.

BOOTLOADING ARDUINO

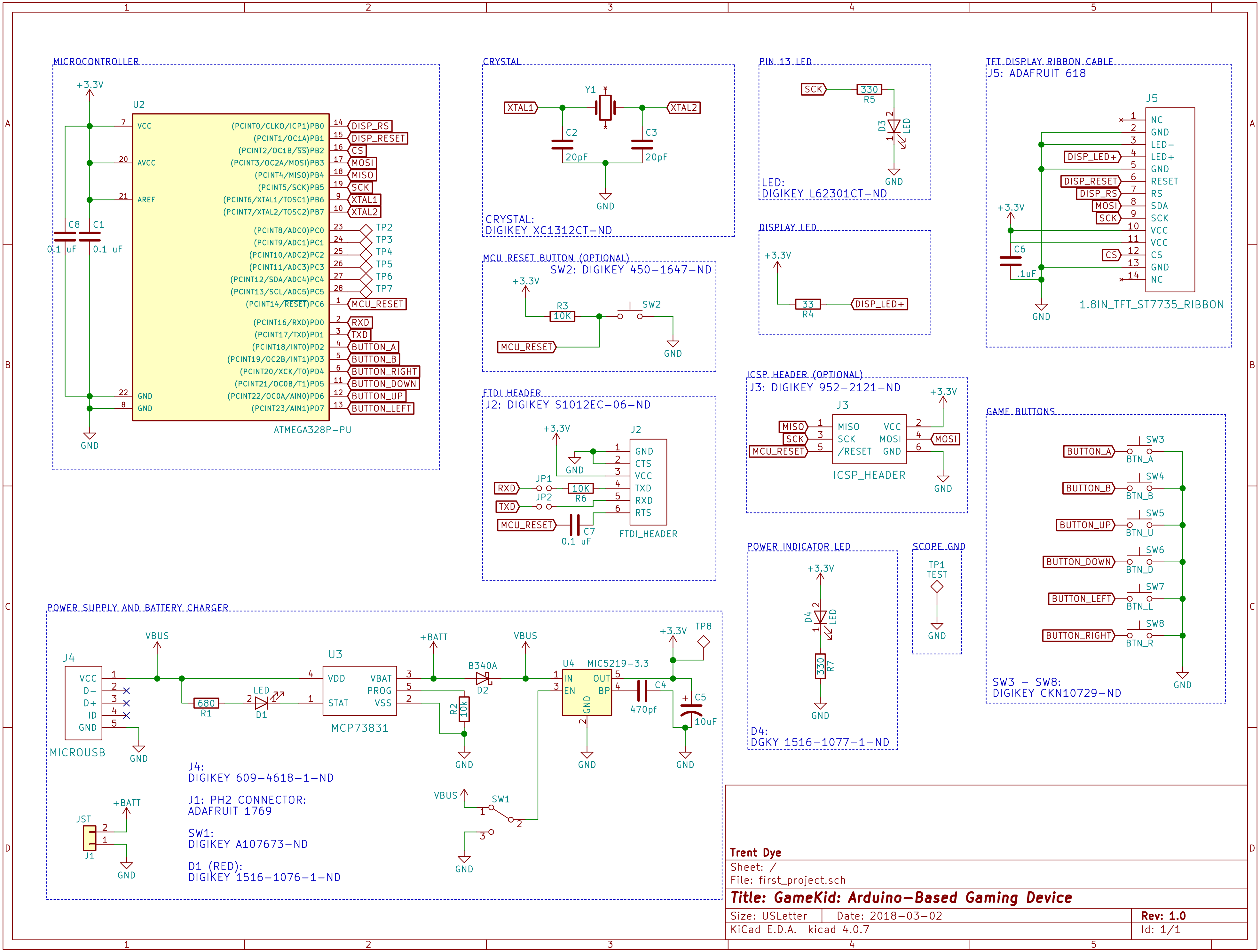
In order to use the GameKid with the Arduino environment, the Arduino bootloader must first be flashed to the ATMega328P chip. Given that the circuit relies on the 3.3V/8MHz "Pro Mini" configuration of the ATMega, which is not typically used with the DIP variant of the chip, pre-bootloaded chips are not widely available online at the time of writing. Additionally, attempts to use the "Arduino [Uno] as ISP" method of installing a bootloader onto the ATMega failed, possibly because the Arduino Uno runs at 16MHz. A verified method of bootloading the ATMega for this application is using an AVRISP Mk. II and Atmel Studio to burn the bootloader onto the chip. With the Arduino IDE installed, the correct bootloader file is located at *C:\Program Files (x86)\Arduino\hardware\arduino\avr\bootloaders\atmega\ATmegaBOOT_168_atmega328_pro_8MHz.hex*.

PROGRAMMING GAMES

Sample game firmware is provided in the Github repository. This firmware contains the digital pin numbers of the buttons and the libraries and function calls necessary to use the LCD screen. The firmware also provides a suggested means of organizing game mechanics for the simple types of games (such as Breakout) that increase their complexity by speeding up the game.

ACKNOWLEDGEMENTS

I would like to thank Stan Reifel, my adviser on this project, for dedicating dozens of hours of his time to discussing everything from big picture to the fine details of the project and for looking over my schematics, PCB routing, and assemblies. I would also like to thank Brad Minch for acting as a third set of eyes for my PCB routing.


MICROCONTROLLER
U2
ATMEGA328P-PU
VCC
AVCC
AREF
GND
+3.3V
C8 0.1 uF
C1 0.1 uF
(PCINT0/CLKO/ICP1)PB0 DISP_RS
(PCINT1/OC1A)PB1 DISP_RESET
(PCINT2/OC1B/SS)PB2 CS
(PCINT3/OC2A/MOSI)PB3 MOSI
(PCINT4/MISO)PB4 MISO
(PCINT5/SCK)PB5 SCK
(PCINT6/XTAL1/TOSC1)PB6 XTAL1
(PCINT7/XTAL2/TOSC2)PB7 XTAL2
(PCINT8/ADC0)PC0 TP2
(PCINT9/ADC1)PC1 TP3
(PCINT10/ADC2)PC2 TP4
(PCINT11/ADC3)PC3 TP5
(PCINT12/SDA/ADC4)PC4 TP6
(PCINT13/SCL/ADC5)PC5 TP7
(PCINT14/RESET)PC6 MCU_RESET
(PCINT16/RXD)PD0 RXD
(PCINT17/TXD)PD1 TXD
(PCINT18/INT0)PD2 BUTTON_A
(PCINT19/OC2B/INT1)PD3 BUTTON_B
(PCINT20/XCK/T0)PD4 BUTTON_RIGHT
(PCINT21/OC0B/T1)PD5 BUTTON_DOWN
(PCINT22/OC0A/AIN0)PD6 BUTTON_UP
(PCINT23/AIN1)PD7 BUTTON_LEFT
CRYSTAL
Y1 XTAL1 XTAL2
C2 20pF
C3 20pF
CRYSTAL:
DIGIKEY XC1312CT-ND
MCU RESET BUTTON (OPTIONAL)
SW2: DIGIKEY 450-1647-ND
R3 10K
MCU_RESET
FTDI HEADER
J2: DIGIKEY S1012EC-06-ND
J2
GND CTS VCC TXD RXD RTS
JP1 JP2
R6 10K
C7 0.1 uF
FTDI_HEADER
PIN 13 LED
SCK R5 330
D3 LED
LED:
DIGIKEY L62301CT-ND
DISPLAY LED
R4 33
DISP_LED+
ICSP HEADER (OPTIONAL)
J3: DIGIKEY 952-2121-ND
J3
MISO SCK /RESET VCC MOSI GND
ICSP_HEADER
TFT DISPLAY RIBBON CABLE
J5: ADAFRUIT 618
J5
NC GND LED- LED+ GND RESET RS SDA SCK VCC VCC CS GND NC
DISP_LED+ DISP_RESET DISP_RS MOSI SCK CS
C6 .1uF
1.8IN_TFT_ST7735_RIBBON
GAME BUTTONS
SW3 BTN_A BUTTON_A
SW4 BTN_B BUTTON_B
SW5 BTN_U BUTTON_UP
SW6 BTN_D BUTTON_DOWN
SW7 BTN_L BUTTON_LEFT
SW8 BTN_R BUTTON_RIGHT
SW3 - SW8:
DIGIKEY CKN10729-ND
POWER INDICATOR LED
D4 LED
R7 330
D4:
DGKY 1516-1077-1-ND
SCOPE GND
TP1 TEST
POWER SUPPLY AND BATTERY CHARGER
J4 VCC D- D+ ID GND
MICROUSB
VBUS
R1 680
D1 LED
U3 MCP73831
VDD STAT VBAT PROG VSS
+BATT
R2 10k
B340A D2
U4 MIC5219-3.3
IN EN GND OUT BP
C4 470pf
C5 10uF
TP8
SW1
JST J1
J4:
DIGIKEY 609-4618-1-ND
J1: PH2 CONNECTOR:
ADAFRUIT 1769
SW1:
DIGIKEY A107673-ND
D1 (RED):
DIGIKEY 1516-1076-1-ND
Trent Dye
Sheet: /
File: first_project.sch
Title: GameKid: Arduino-Based Gaming Device
Size: USLetter
Date: 2018-03-02
Rev: 1.0
KiCad E.D.A. kicad 4.0.7
Id: 1/1

Bill of Materials – Schematic Components

Identifier(s)	Value	Quantity	Supplier	Part Name	Supplier P/N
R1	680 Ohms	1	Digikey	RES SMD 680 OHM 5% 1/4W 1206	311-680ERCT-ND
R2, R3, R6	10K Ohms	3	Digikey	RES SMD 10K OHM 1% 1/4W 1206	311-10.0KFRCT-ND
R4	33 Ohms	1	Digikey	RES SMD 33 OHM 5% 1/4W 1206	311-33ERCT-ND
R5, R7	330 Ohms	2	Digikey	RES SMD 330 OHM 1% 1/4W 1206	311-330FRCT-ND
C1, C6, C7, C8	0.1 uF	4	Digikey	CAP CER 0.1UF 50V X7R 1206	311-1179-1-ND
C2, C3	20 pF	2	Digikey	CAP CER 20PF 50V C0G/NPO 1206	311-1153-1-ND
C4	470 pF	1	Digikey	CAP CER 470PF 50V X7R 1206	1276-2778-1-ND
C5	10uF	1	Digikey	CAP TANT 10UF 10V 10% 1206	478-1654-1-ND
Y1	8MHz	1	Digikey	CRYSTAL 8.0000MHZ 16PF SMD	XC1312CT-ND
J1	-	1	Adafruit	JST-PH 2-Pin SMT Right Angle Connector	1769
J2	-	1	Digikey	CONN HEADER .100" SNGL STR 6POS	S1012EC-06-ND
J3	-	1	Digikey	DIL VERTICAL PC TAIL PIN HEADER	952-2121-ND
J4	-	1	Digikey	CONN USB MICRO B RECPT SMT R/A	609-4618-1-ND
J5	-	1	Adafruit	1.8" SPI TFT display, 160x128 18-bit color - ST7735R driver	618
U2	-	1	Digikey	IC MCU 8BIT 32KB FLASH 28DIP	ATMEGA328P-PN-ND
U3	-	1	Digikey	IC CONTROLLR LI-ION 4.2V SOT23-5	MCP73831T-2ACI/OTCT-ND
U4	-	1	Digikey	IC REG LINEAR 3.3V 500MA SOT23-5	576-1281-1-ND
JP1, JP2	-	1	Digikey	RES SMD 0 OHM JUMPER 1/4W 1206	311-0.0ERCT-ND
D1	-	1	Digikey	LED RED CLEAR 2SMD R/A	1516-1076-1-ND
D2	-	1	Digikey	DIODE SCHOTTKY 40V 3A SMA	B340A-FDICT-ND
D3	-	1	Digikey	LED RED DIFFUSED 1206 SMD	L62301CT-ND
D4	-	1	Digikey	LED YELLOW CLEAR 2SMD R/A	1516-1077-1-ND
SW1	-	1	Digikey	SWITCH SLIDE SPDT 200MA 30V	A107673-ND

Bill of Materials – Additional Components

Name	Qty	Supplier	Supplier P/N
Case Bottom - 3D Print	1	In-house	N/A
Case Top - 3D Print	1	In-house	N/A
Button Directional - 3D Print	4	In-house	N/A
Button Nondirectional - 3D Print	2	In-house	N/A
Brass Screw-to-Expand Inserts for Plastics, M3 x 0.5 mm Thread Size	4 (Pack of 50)	McMaster-Carr	94510A030
Black-Oxide Alloy Steel Socket Head Screw, M3 x 0.5 mm Thread, 16 mm Long	4 (Pack of 100)	McMaster-Carr	91290A120
Clear Acrylic Sheet, 12" x 12" x 1/8"	1	McMaster-Carr	8589K41
Kapton® Polyimide Masking Tape	1	McMaster-Carr	7648A711
Light Duty Foam Tape - Adhesive on Both Sides	1	McMaster-Carr	7598A915
Gaffer's or other opaque tape	-	-	-
Hot glue	-	-	-