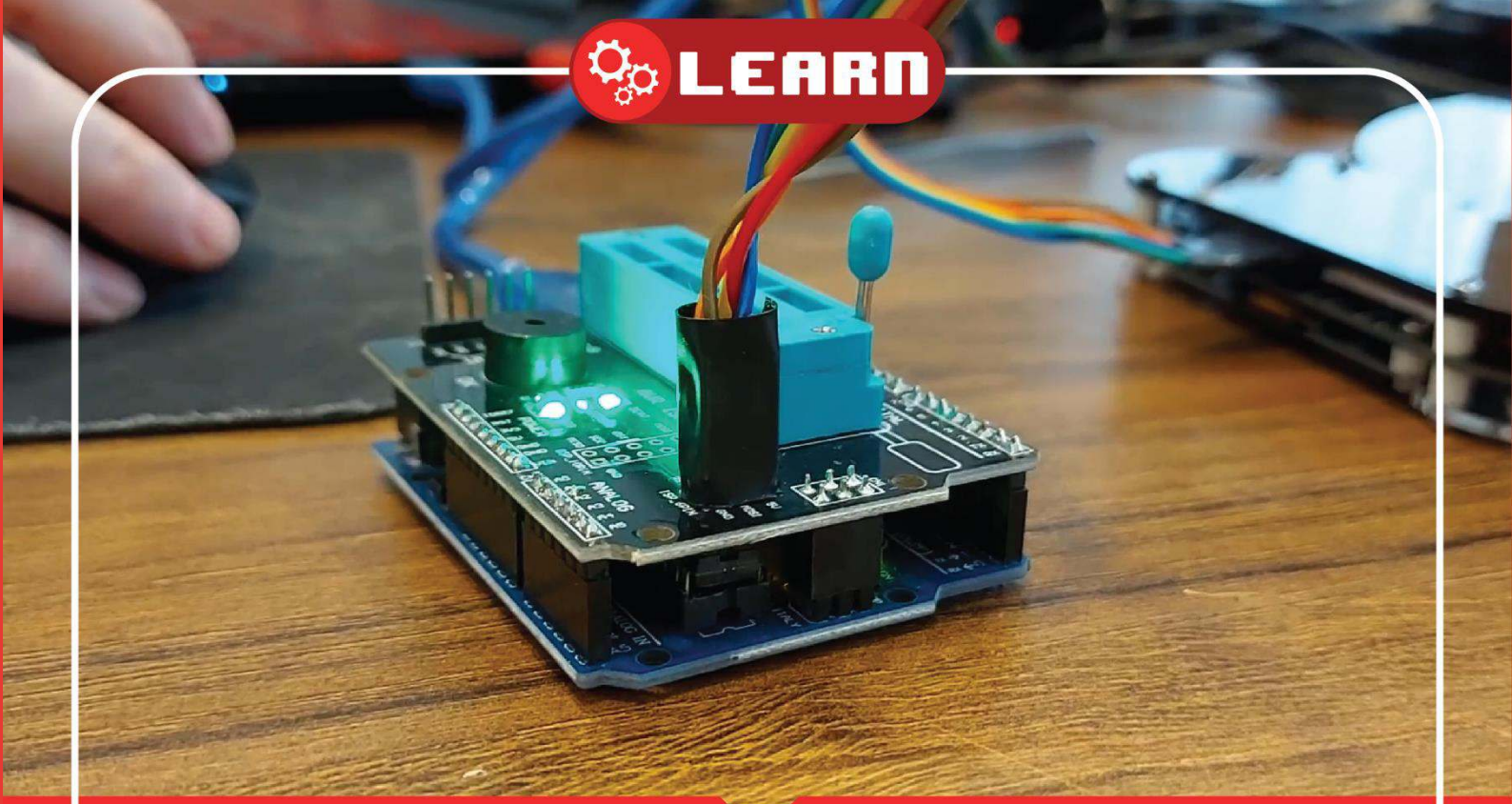
LEARN



Uploading the Bootloader

With 8BitCADE

Compatible with:

8BitCADE XL

Contents

Written by the 8BitCADE Team

Support@8bitcade.com

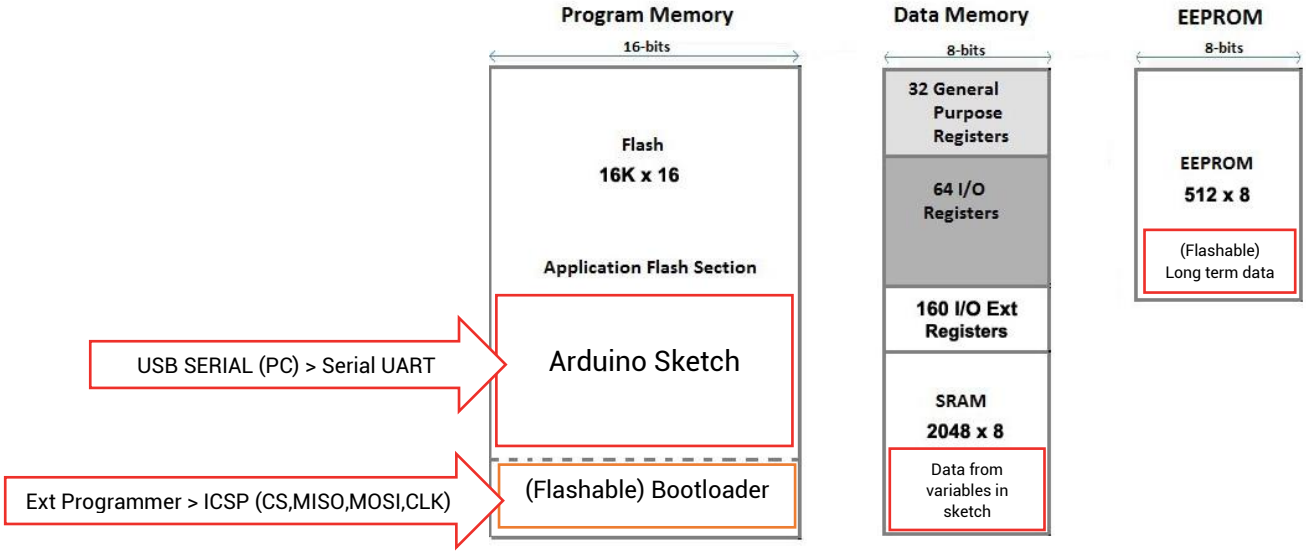
Version 1

ATmega 32U4 has three types of memory

The chip contains 32 kilobytes of internal flash memory, 1 kilobyte of EEPROM and 2.5 kilobytes of SRAM. The flash and EEPROM memory are rewritable electronically, using the Arduino IDE. The information is written to the memory and remains (called volatile) after the chip is powered down (switched off) but the SRAM is a memory which only saves information while power is supplied (called non-volatile) and when the power removed all the information is erased. The program memory contains your sketch you loaded via the Arduino IDE and the bootloader, which is a short, protected program that runs when you turn the chip on, or press the reset button. Its main function is to wait for the Arduino software on your computer to send it a new program which it then writes to memory. The bootloader is an important sketch and is positioned at the end of Program Memory to protect it and can only be rewritten using ICSP (In Circuit Serial Programming).

When we refer to "boot loading" the **ATmega 32U4** chip, we are talking about using a special device (called an In-Circuit-Serial-Programmer or ICSP) to replace the bootloader software.

Overview of the Different Types of Memory of an AVR Chip

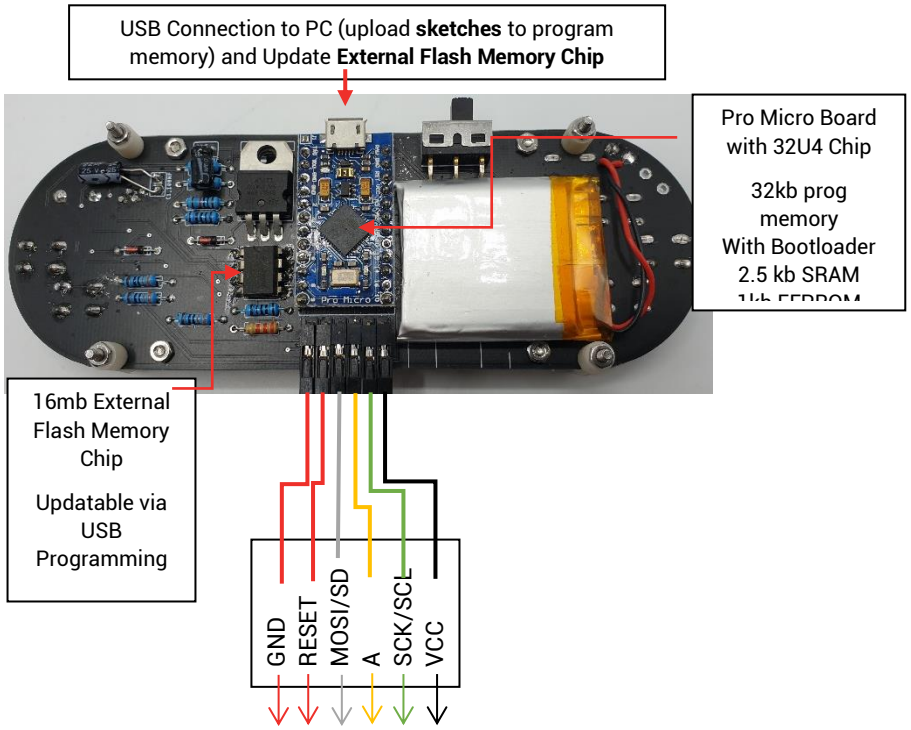


Program Memory - 32KB nonvolatile memory. This is used for storing your **sketch** (program) and the bootloader. **This code remains in the memory of the chip even if the power is removed from the circuit.** The program or actual sketch code is loaded via an FTDI chip (a UART Bridge) converting your PC USB serial to TTL serial the AVR chip can read.

Boot Flash Section (Bootloader) enables you to program the Arduino board easily using a PC attached via a USB connector. Its protected and is only accessible via special pins (SPI) if you want to reprogram it using ICSP.

SRAM Memory -2.5 KB volatile memory. This is used for storing data from your sketch variables while the program is running. Once the power is turned off this data is lost. A good example of the type of data might be the position of a player on the screen which needs to be tracked so the position can be updated, like the ball in the game 'Pong'.

EEPROM Memory -1KB nonvolatile memory. This can be used to store data that must be available even after the board is powered down and then powered up again. A good example of data stored here would be high scores of a game, small amount of data which would need to be saved to be displayed for later use.



ICSP Programmer 6 Pin Connector
(Access Bootloader in Prog Memory)

Re-Programming the Bootloader

Introduction

There are two ways to program the main microcontroller (ATmega 32u4 chip) on the 8BitCADE XL. One is to reprogram the entire chip using a specialist hardware called an AVR ICSP Programmer. The other is to use a **bootloader** that is pre-programmed onto the main microcontroller that allows the chip to re-program itself.

The bootloader is basically a **.hex** file that runs when you turn on the microcontroller. It is very similar to the BIOS that runs on your PC. It does two things. First, it looks to see if the computer is trying to program it. If it is, it takes the program from the computer and uploads it into the ICs memory (in a specific location so as not to overwrite the bootloader). This is why when you try to upload code, the Arduino IDE resets the chip. This basically turns the chip off and back on again so the bootloader can start running again. If the computer isn't trying to upload code, it tells the chip to run the code that's already stored in memory. Once it locates and runs your program, the Arduino continuously loops through the program and does so as long as the board has power.

Programming the chip with an **updated** bootloader will add access to a **menu system** on the external serial flash memory chip which can store up to 500 games. Arduino chips (AVR chips) like the ATmega 32u4 chip on the Pro Micro, generally have the bootloader added at the factory.

Its protected so that new users or beginners don't overwrite it by mistake while programming the chips using software like the Arduino IDE. To reprogram it with a new bootloader we need to use the SPI protocol, SPI connections and an AVR programmer like a USB ASP (USB Tiny) or ICSP Programmer (like the DIYMORE Shield).

Procedure

MR.Blinky created the Arduboy-homemade-package for homemade Arduboy (we use this to program our 8BitCADE XL). His package includes the **board drivers and library of Arduboy that works with different versions of the original Arduboy like the 8BitCADE XL**.

To achieve this you need 2 things:

1. **Correct set-up** in the **Arduino IDE** for the Bootloader
2. **Correct connections** between an ICSP Programmer and the 8BitCADE giving access to the bootloader memory inside the ATmega 32u4 chip on the main microcontroller.

(See video in learn section for a visual guide)

Correct Set-up for Arduino IDE

1. Browse to MR.Blinky's GitHub folder for the homemade Arduboy.

https://github.com/MrBlinky/Arduboy-homemade-package

2. Follow the instructions on the GitHub to configure your Arduino IDE with the homemade package **or follow the instructions below.**

3. First **copy** the url of the "Additional board manager" for Arduboy homemade package.

https://raw.githubusercontent.com/MrBlinky/Arduboy-homemade-package/master/package_arduboy_homemade_index.json

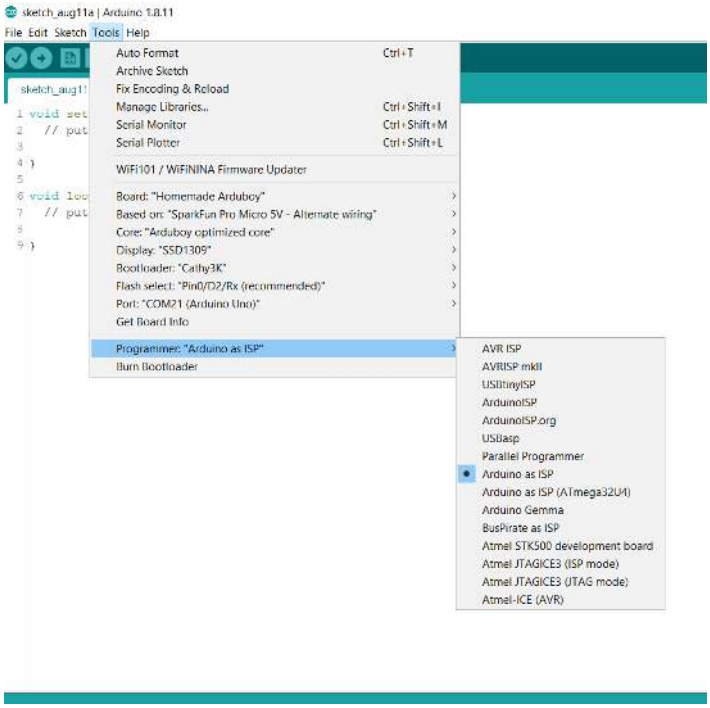
4. Start Arduino IDE. Click **Preferences** from the Arduino top menu. **Paste** this text into the **"Additional Boards Manager URLs"** Note: If you already have other text on this field, insert this additional text at the beginning, then add a ", " and keep the other text intact.

5. Exit Arduino IDE and start the IDE again to take effect of the change above.

6. Click Tools > Board: Board Manager. Enter homemade to search. Select to install the Arduboy homemade package by Mr.Blinky. Then click update to get the latest version. The package will be added to Arduino.

7. Now select Tools>Board: "Home-made Arduboy" and select the following parameters for Homemade Arduboy based on:

Board: "SparkFun Pro Micro 5V" - "alternative wiring"
Core: "Arduboy Optimized core"
Bootloader: "Cathy3K"
Programmer: USBasp or Arduino as ISP (if using DIYMORE Shield)



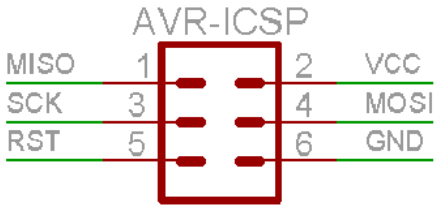
8. **WARNING: Turn off the 8BitCADE – the unit must only be powered by your** AVR ICSP Programmer. Attached the cables from the AVR Programmer (we use DIYMORE Shield) ICSP_6Pin connector or other AVR ICSP Programmer and connect them as follows:

Correct Connections Between AVR ICSP Programmer & 8BitCADE XL with Explanation

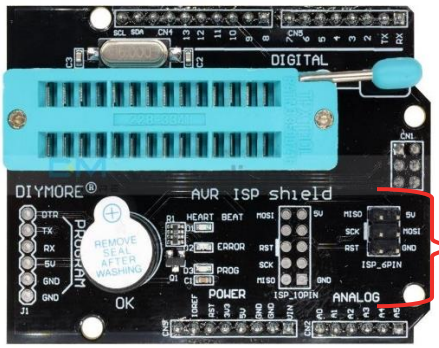
As previously mentioned, to access the bootloader section of your program memory of your 8BitCADE XL, we need to use an AVR 'In Circuit Serial Programmer'. This will help you get access to all the **program memory** of your 8BitCADE Microcontroller (Pro Micro with the 32u4 chip) including the bootloader. This programming can actually be done in a number of different ways, although the principles are the same, we need to use another microcontroller to use SPI to program the bootloader on the microcontroller inside the 8BitCADE XL. At 8BitCADE, we use an AVR In-Circuit-Serial-Programmer by DIYMORE as we can use this for other operations like programming ATmega chips!

The ICSP (In-Circuit Serial Programming) header pins connects the SPI (Serial Peripheral Interface) bus and we use the SPI protocol to transfer the data from the Arduino IDE on your PC to the 8BitCADE via the ICSP Programmer. See the **Make Guide**, part dictionary for information on SPI. The pins for the ICSP 6PIN header are as follows:

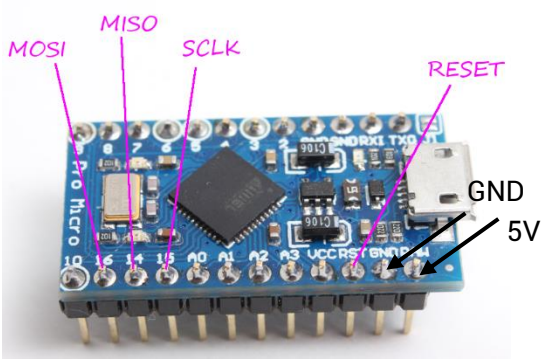
1 - MISO - Master Input, Slave Output - output from slave to master
2 - VCC - 5V
3 - SCK - Serial Clock - keeps the communicated data in sync
4 - MOSI - Master Output, Slave Input - output from master to slave
5 - RST - Reset
6 - GND - Ground



In-Circuit-Serial-Programming (ICSP)



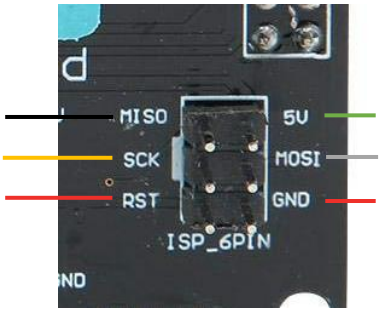
ICSP connections to access bootloader of 8BitCADE XL



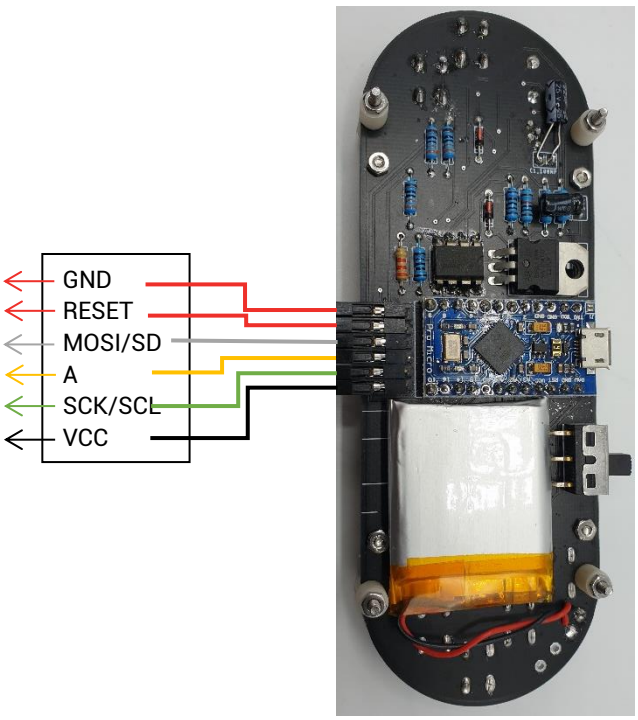
Connections on Microcontroller

The connections on the microcontroller have been broken out to the main ICSP connector on the 8BitCADE XL Board for easy connection. See below.

How to connect to the 8BitCADE XL using the ICSP Connector



ICSP Programmer 6 Pin Connector



09. Connect AVR ICSP Programmer (DIYMORE Shield) to your PC via the USB connection. This will connect the ICSP Programmer to the Arduino IDE.

10. Your 8BitCADE should be powered on now through DIYMORE Shield ISP_6Pin connector or other ICSP Programmer.

11. Click the Tools> Boards>Burn Bootloader button on the Arduino IDE.

12. Check the message to see if the bootloader burn is successful.

13. If not, check the cable and make sure your connections are correct.

14. If successful, the 8BitCADE XL will show the following screen: **USB Boot Logo (see picture) all you need to do now is disconnect the AVR Programmer, turn the unit on and it should reboot to the 8BitCADE Splash Screen.**

USB Boot Logo in the middle of the screen shows you have successfully programmed the boot loader. If the serial flash chip is also loaded then disconnect the AVR Programmer and press reset and the 8BitCADE Loader Screen should appear, giving you full access to the games image (200 games).