synthiam.com

Connecting Arduino to ARC

Program an Arduino and connect it to ARC to become an EZ-B.

Last Updated: 8/29/2021

Compatible Arduinos

This tutorial currently covers what Arduino's are compatible with ARC, where to download the firmware, how to flash the Arduino and connect to ARC.
Once you follow this tutorial and program an Arduino, you will be able to connect an Arduino to ARC and control the Arduino. Essentially, you will be converting an Arduino into an EZB. The best part about this is having access to the source code means you can add new functionalities to the Arduino and extend the capabilities of ARC. The one item to note is each Arduino will have different capabilities or lacking capabilities (i.e. video, audio, etc.).

ARC Version

- This feature works with ARC 2019.01.31.00 Beta or newer

Arduino Capabilities

Additionally, each Arduino model will have a different MCU (microcontroller) with different capabilities for servos, PWM, and Analog. Consult the datasheet/manual of the Arduino you're using to ensure you understand what ports are available for different commands. For example, some Arduino models only allow PWM/Servos on specific digital ports. Meanwhile, other Arduino's share I2C with Analog ports. You will have to determine what ports are available because we have no idea due to the number of Arduino versions out there. For example, servo ports on Arduino UNO must be PWM ports 3, 5, 6, 9, 10, and 11. These limitations are that of the Arduino libraries and hardware.
There's a significant number of Arduino's variations out there each with slight modifications. This means the base firmware provided in this tutorial can be modified for various Arduino configurations. Also, you are free to add new features to the Arduino firmware and register them as Capabilities to be used in ARC. But, that's a future edit of this tutorial
.

*Arduino Mega showed in the picture above

Compatible Official Arduinos

The ARC software is compatible with a number of Arduino products, which the firmware can be found on their respective pages. Find your Arduino board and download the firmware from this page: https://synthiam.com/Support/Hardware/Hardware-Overview
Download the firmware source code for your Arduino version or compatible from the above link and continue to the next step of this tutorial. *Note: compatible or clone Arduino's will require a toolset and library installation to compile the firmware. Check the Arduino compatible product manual for instructions on adding libraries and tools to the Arduino IDE for clone hardware.

1. On the previous step, you were instructed to download a firmware soucecode zip file for the Arduino that you'd like to use. Extract the contents of your zip file somewhere on your hard drive. I create a Arduino folder in My Documents and keep them in there.

2. Enter the folder with your Arduino firmware source file. I'll use the Arduino Mega as an example, but the process is the same for the other Arduino versions.

3. Do you have Arduino IDE programming software installed? If so, double click on the *.INO file and it will load the project into the Arduino IDE software. Otherwise, download Arduino software from HERE.

4. When the software loads, the main file will be the EZ-B firmware. The only field that really matters for you to notice is the BAUD_RATE. You will need to know the BAUD_RATE when connecting to EZ-Builder in the next step of this tutorial.

5. Connect your Arduino to the PC via the USB cable

6. Let's configure the Arduino IDE for your Arduino board. In my example, I'm using the MEGA so I selected that. If you're using an UNO or Duemilanove, select that instead.

7. Specify the COM PORT for your Arduino to be programmed

8. Press the UPLOAD button, which will compile and program the Arduino to become an EZ-B

9. There you go, the Arduino has been programmed and is now an EZ-B. The next step will demonstrate how to connect ARC to the EZ-Arduino. Or is it EasyDuino? OR EZDuino?

Remember, you don't have to perform these steps each time. The Arduino will remember the programming when it is turned off and back on again. The program is inside of the Arduino memory until it is reprogrammed. So, feel free to close the Arduino IDE program and move to the next step.

Connecting to ARC

On the previous step, we programmed the Arduino to become an EZ-B. Now that the code is running on the chip, we can begin using it with ARC.

1. Load ARC and press the CONFIG button on the connection control

2. Press the SETTINGS tab and select the BAUD RATE for the EZB Index that will be your Arduino. If you're connecting to multiple devices (i.e. EZ-B v4, IoTiny and Arduino), you will want to have each device on a different connection index. I'm using connection index 0 (zero) in my example. Change the BAUD RATE for the connection index that you wish to use. The BAUD RATE was specified in the previous step. By default, most of the EZ-Arduino firmwares will be 57600 baud.

3. Press SAVE

4. Click on the ADDRESS input of the device index that you configured for the Arduino. In my example, I'm using index 0 (zero). The ADDRESS will display a number of detected addresses and devices, including a COMx port, which is the Arduino. Select the COMx port. My Arduino is on COM6.

5. Now press the CONNECT button and EZ-Builder will connect to the Arduino. The capabilities of the particular Arduino device are detected by EZ-Builder and displayed in the debug window. In the example below, I added a debug window to demonstrate the connection log.

Notice the capabilities detected by the Firmware used on this Arduino. There are a number of capabilities listed, but there's a larger number of capabilities missing. For example, there is no Video or Audio support. Consult the capability list of the firmware so you understand what features may be missing before using. If a feature is missing from the firmware, ARC will notify you when attempting to use the capability.

You may now begin using your Arduino with ARC. The port numbers align with the ports of the Arduino. Based on the Arduino hardware version, you may be limited to ports. For example, the Arduino UNO has 14 digital ports and only 2-13 are usable. This is because 0 and 1 are used for the USB. If you select a port higher than 13 in ARC, nothing will happen on the Arduino because there's no port. The same applies to Analog ports, as the UNO only has 6 ADC ports.
The Arduino Mega, however, has enough ports to satisfy ARC’s port selection dialog.

Customizing Firmware

The firmware can be modified and optimized for additional features. When the firmware is modified, the new capabilities must be registered with Synthiam website.
I should add a disclaimer that this tutorial is not complete. You can customize the EZ-Arduino firmware with new features (called capabilities) that EZ-Builder will recognize, although that part of this tutorial hasn't been completed yet.