

# GoJimmyPi

Sunday, December 9, 2018

## SWD Debugging the Arduino MKR-WiFi-1010 with Atmel ICE

Over the weekend I received my new [Arduino MKR-WiFi-1010](#) (the ABX00023) that I purchased from [Amazon](#), as seen here along with my [Atmel ICE](#):



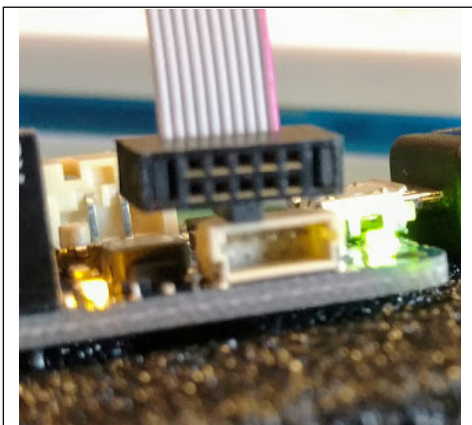
Warning: after writing this blog I realized that using the Atmel ICE can end up causing problems with the bootloader when later attempting to upload a sketch via the serial port! This was [easily resolved](#) (everything is easy, when you know how), but beware.

Ironically, I found this board shortly after commenting to [Peter Scargil](#) that [I really didn't think anything could replace the Espressif ESP8266 \(other than the ESP32\)](#) since it has been around for so long and has such a strong community and momentum. I found the 1010 shortly after seeing a [tweet from MicroChip regarding the ATmega4809](#). (an 8-bit Arduino, 16MHz ceramic resonator, WiFi board using the ECC608 Crypto-Authentication chip for \$44.90). The MKR1010 (32 bit!) was listed as "related product". Now, I knew the MKR-1010 board had been released some time ago - but at the time I seem to think it was rather expensive and didn't think much more about it. However at \$29.90 - that's just too appealing to pass up. Amazon prime to the rescue for me gizmo addiction: delivered the next day at no extra change! Thank you Prime!

It has been some time since my last blog post. I have a draft blog of some things that I learned with [my new CNC](#), but it is rather messy at the moment, so instead this weekend I am back to debuggers and this MKR-1010 board.

One of the main points of interest with this board - is that in addition to the SAMD21 (specifically the ATSAMD21G18A) it *also* has an on-board ESP32 ... *and* the MicroChip ECC508 Crypto-Authentication chip! Although I really like the ESP8266 boards, I will say that JTAG debugging was not exactly the easiest and most robust, as noted [here](#) and [here](#). One thing about the Atmel ICE is that in Atmel Studio, it is a fairly robust and reliable hardware debugger.

So fresh out of the gate, I was [disappointed that I did not have a cable](#) for (what I thought) was the Single Wire Debug (aka SWD) connector:



### gojimmypi links

Follow [@gojimmypi](#)

[@gojimmypi](#) on GitHub

[@gojimmypi's Gists](#)

Favorite Gizmos

### Blog Archive

► 2019 (16)

▼ 2018 (17)

▼ December (2)

Favorite Gizmos of 2018

SWD Debugging the Arduino MKR-WiFi-1010 with Atmel...

► September (1)

► July (1)

► June (1)

► May (1)

► April (2)

► March (6)

► February (2)

► January (1)

► 2017 (27)

► 2015 (3)

► 2014 (7)

### Labels

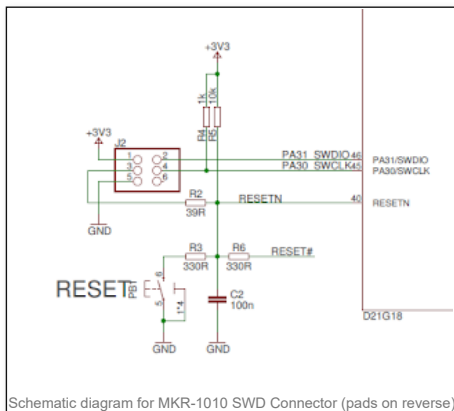
- 32u4
- 8051
- Adafruit
- Adafruit.IO
- ALINX
- Altera
- Arduino
- ARM-USB-OCD-H
- ATMEGA328
- Atmel
- Atmel ICE
- Atmel Studio
- AVR
- BlackMagic
- Blue Pill
- Build\_ESP8266
- CP/M
- Crowd Supply
- CubicSDR
- Cyclone IV
- cygwin
- Debug
- Debugger
- DevKitC
- dfu-util
- dps5005
- dps5015
- ECP
- elecrow
- Embedded Debugging
- EP4CE6E22C8
- ESP32
- ESP32 WROOM-32
- ESP32\_Core\_Board\_V2

Comparing a 50 mil connector to what I thought was the SWD connector.

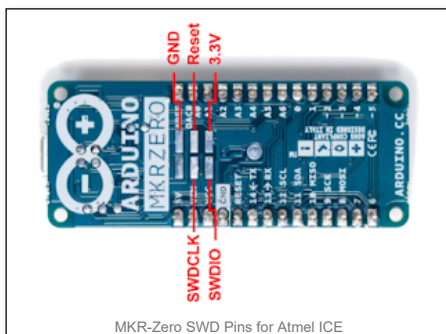
Note that black connector not a 0.1 pin spacing, it is already the small 50 mil connector that is on the Atmel ICE. The white connector below *is even smaller* than that!

I spent an embarrassingly long time scouring the internet for anything I could find on "Arduino MKR 1010 SWD Debugging Connector". I could not even determine what that tiny white connector is called so that I could order one. I even [posted a plea for help](#) on the [Arduino MKRWiFi1010 forum](#),

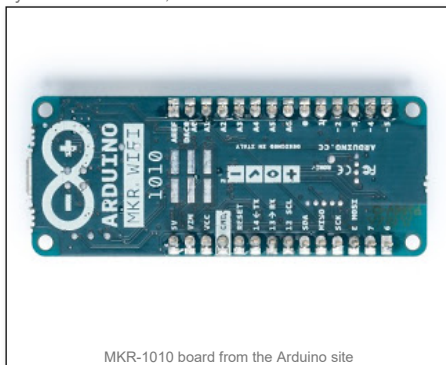
Eventually, I found [this post](#) where someone was asking about connecting the Atmel ICE SWD to the MKR-ZERO. Although I had no indication the 6 pads on the underside of my MKR-1010 were actually the J2 SWD pins, I was hopeful that the folks at Arduino would have standards and consistency.



The Maker Zero SWD pins are as shown here, in this diagram provided by [Federico\\_Vanzati](#):



Keep in mind I had no experience, no knowledge of the MKR-Zero, however the MKR WiFi 101 is just a bit similar, once you see them both, eh?

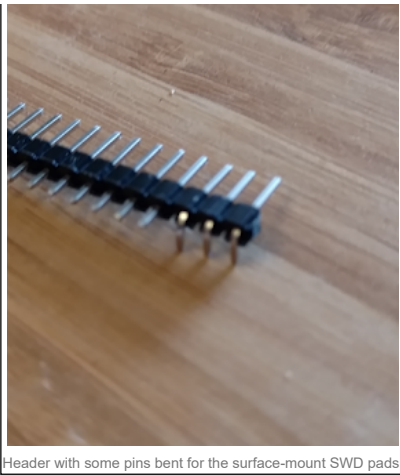


So although I was confident and hopeful that I could finally get the Atmel ICE to connect, I was still stuck with 6 pads and no holes. I really wish the Arduino folks would have made a more debug-friendly connector.

Fortunately, I had some of those snap-apart headers on hand. I bent two sets of 3 pins...

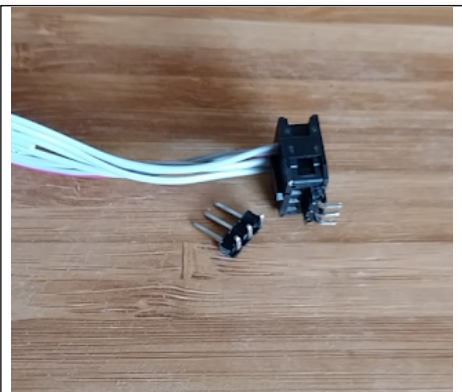


- ESP8266
- Espressif
- Flow Sensor
- Forth
- FPGA
- FTDI
- ftx-prog
- GDB
- GitHub
- Gizmo
- GPIO
- GUI
- hackaday
- Heur.AdvML.B
- hidden ssid
- icestorm
- icestudio
- ISP
- JTAG
- JTAG. ISE
- libopenm3
- LoRa
- M5Stack
- Mimas V2
- MinGW
- Mkr-WiFi-1010
- MKR1010
- Mouser
- MQTT
- MyForth
- NESDR
- nextpnr
- node
- Nooelec
- Norton
- npm
- Numato
- Olimex
- OpenOCD
- OTA
- Pano Logic
- Pressure Sensor
- Probe
- Programming
- Quartus
- RadioHead
- raspberry pi
- Remote Control
- RFM95
- RISC-V
- RISC-V
- RTL2832U
- SDR
- Segger
- Serial Port
- sino:bit
- Somfy
- Spartan6
- SPI
- SPIDriver
- STM Smart V2 board
- STM32
- STM32f103C8T6
- SWD
- sysprogs
- tinyFPGA
- Ubuntu
- uipro
- ULX3S
- USB
- USB Blaster
- verilog
- VHDL
- vhd2vl
- Visual Studio



Header with some pins bent for the surface-mount SWD pads.

...and stuck them both into a 6-pin cable connector to hold them in the proper position for soldering:

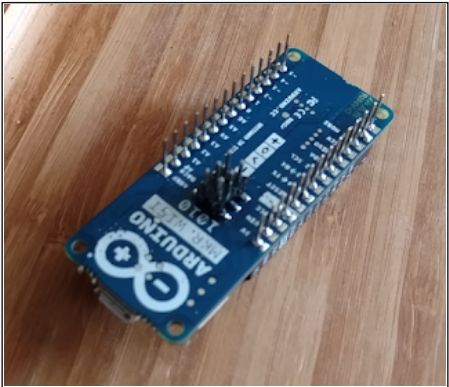


Preparing the home-made surface mount headers in a 6 pin connector.

Be sure that the longer header pin is not pulled through when bending them: Not only do you want to have a good length for electrical contact, but you also don't want them too long on the PCB. Yes, there's a solder mask layer that should act as an insulator, but I would not count on it. Make sure that pins are only as long as the solder pads. I needed to trim mine.

Note that using sets of three will increase overall mechanical strength - but as they are surface mounted, care should be taken when inserting & removing connections.

So the reality of annoying future breadboard use becomes quite apparent once the header is soldered in place. Clearly the designer prioritized the Arduino logo over the practicality of pad header placement:



My MKR-WiFi-1010 SWD Header Connection, freshly soldered on

I'm thinking if I plug it into a breadboard, I'll just use some of those stackable headers. Still, this will leave many more mechanical connections and many more places for connection problems.

From the MicroChip / Atmel ICE docs - these are the pins used for SWD on the SAMD21 chips:



- [VisualGDB](#)
- [VisualMicro](#)
- [WeMOS](#)
- [WiFi](#)
- [WSL](#)
- [X-Windows](#)
- [Xilinx](#)
- [yosys](#)
- [Z80](#)

Report Abuse

- [Home](#)

Search This Blog

Search

Connection to a custom 100-mil header

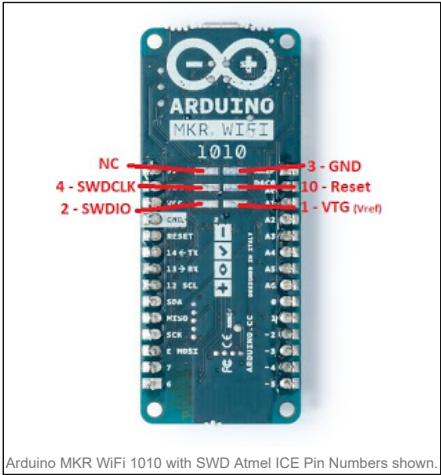
The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR or SAM connector port and the target board. Six connections are required, as described in the table below.

Table 3-7. Atmel-ICE SWD Pin Mapping

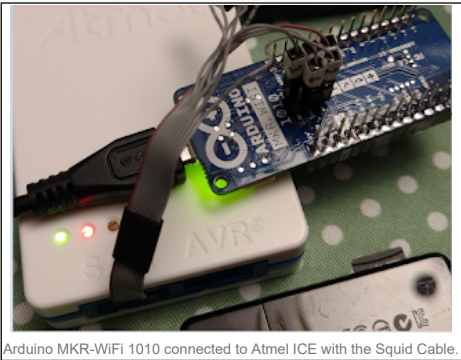
| Name   | AVR port pin | SAM port pin | Description  |
|--------|--------------|--------------|--|
| SWDCLK | 1            | 4            | Serial Wire Debug Clock.                                       |
| SWDIO  | 5            | 2            | Serial Wire Debug Data Input/Output.                           |
| SWDIO  | 3            | 6            | Serial Wire Output (optional- not implemented on all devices). |
| nSRST  | 6            | 10           | Reset.   |
| VTG    | 4            | 1            | Target voltage reference.                                      |
| GND    | 2, 10        | 3, 5, 9      | Ground.  |

SAMD Pins for SWD Debugging with Atmel ICE (from the User Guide)

Upon reading the pinouts and looking at the cables and adapters that I have for the Atmel ICE, I realized that my only option was to use that handy-but-annoying Squid Cable.

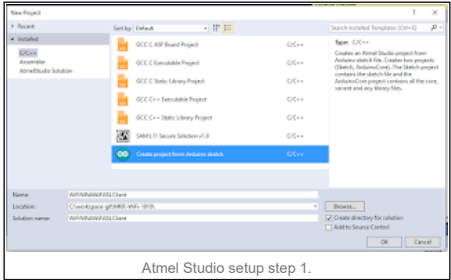


Here's what the board looks like connected to the Atmel ICE:



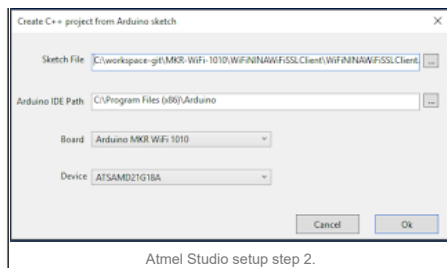
I sure wish a simple 10-pin header would have been installed on the MKR boards - that would connect directly to the Atmel ICE debugger. Well, no worries.. good to go!

There's a MKR-WiFi-1010 tutorial on actually using the WiFi [here on GitHub](#). I actually used the [WiFi tutorial page](#) to get started. I took the ino file and pasted it into the Arduino IDE and compiled to ensure everything was working properly. Then I created a new Atmel Studio project from the ino.



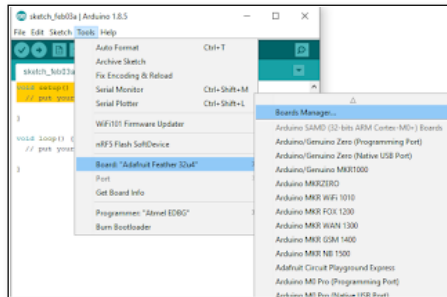
And on the next screen, select the location of the ino project from the Arduino IDE:



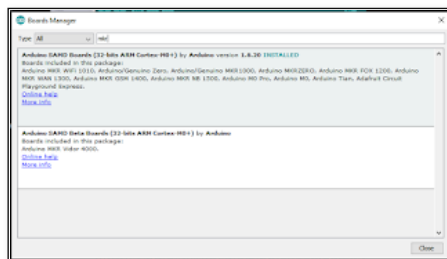


Atmel Studio setup step 2.

Edit Feb 2019: If you don't see the board listed, then launch the Arduino IDE and ensure it is installed via Tools - Board - Board Manager:

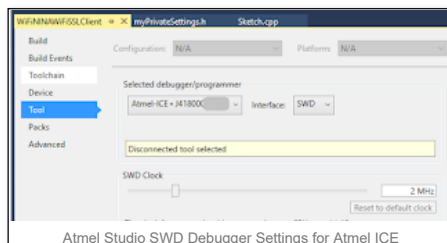


Type "MKR" to shorten the list:



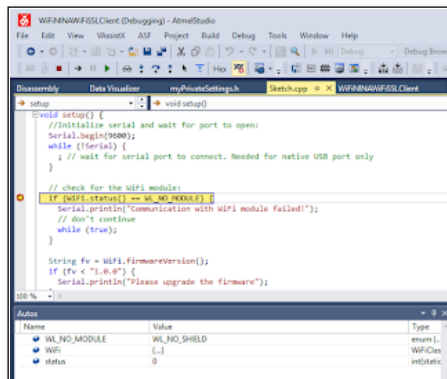
/Edit... back to Atmel Studio:

Connection in Atmel Studio is pretty straightforward, just select the Atmel ICE and ensure "SWD" is used:



Atmel Studio SWD Debugger Settings for Atmel ICE

After that... voila! Click the Debug menu and step in, set breakpoints, and more!

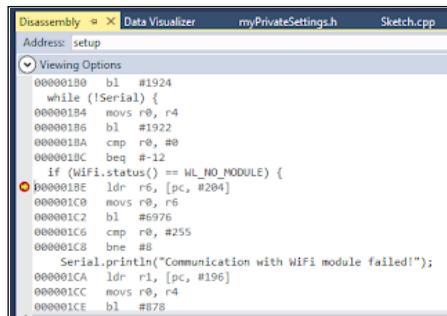


One thing that is not super obvious when debugging with the Atmel ICE, is most sample code is "wait forever until serial port available" during setup. Well, the Atmel ICE is not the serial port!

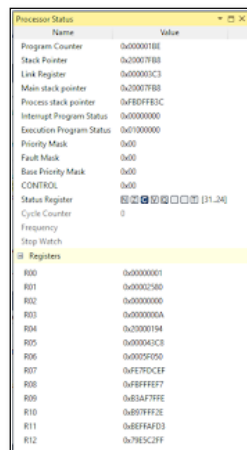
```
//Initialize serial and wait for port to open:
Serial.begin(9600);
while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
}
```

So be sure to connect something to the serial port! The serial monitor in Atmel Studio is a bit wonky. I simply use [putty](#).

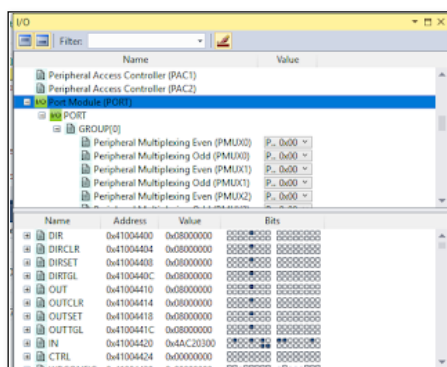
See the Disassembly tab to view the assembly language generated from the source code:



The Processor button show interesting information:



Also cool to browse and inspect I/O registers:



This is SO much better than a zillion serial.println statements!

Well, I hope this saves you the hassle I encountered when trying to use the Atmel ICE hardware debugger with the Arduino MKR-WiFi 1010.

I've added a [MKR-WiFi-1010 GitHub Repository](#) with the source code and relevant pictures and documentation.

So ya, perhaps this Arduino will take over the Espressif chips ... but as there's an Espressif on-board this guy, perhaps that's not a fair comparison. Still - it is a quite interesting board at a somewhat reasonable price.

Resources, Inspiration, Credits, and Other Links:

- [WiFinINA library](#)



- [Adafruit's Proper debugging of ATSAM21 Processors - Restoring Bootloader](#)
- [Bootloaders at ArduinoCore-samd/bootloaders/mkrwifi1010/](#)
- [MKR1000 Firmware update by badballscrewbob](#)

Please leave comments, ideas, suggestions below (moderated, sometimes delayed) or send me a message at gmail, or find [me on twitter](#).

By [gojimmypi](#) at [December 09, 2018](#)

Labels: [Arduino](#), [Atmel ICE](#), [Atmel Studio](#), [Debugger](#), [Embedded Debugging](#), [Mkr-WiFi-1010](#), [MKR1010](#), [SWD](#)

## 8 comments:



**Marcio Teixeira Jr.** January 9, 2019 at 1:07 PM

Congrats on your blog, gojimmypi. Very interesting and informative. I ran through quite the same problem with my new MKR WAN 1300 board. I completely agree that this SWD interface on the MKR boards are very NON\_FRIENDLY and not properly documented. I would like to add the information that the ATMEEL-ICE basic kit that comes with a 6 pin, 100mil headed cable CAN'T be used on the MKR boards. The signals are completely different. Right now I'm using the VSCode/PIO interface but I would like to test the AtmelStudio with the Arduino. I installed it, but when trying to import the .ino project the interface only shows a few boards (like Uno, Nano, Leonard, Mega), mostly atmega32 processors. Can you tell me how did you added the support for the MKR boards, as you show at your pictures here? Everywhere I checked they always shows examples using Arduino UNO.

Thanks.

[Reply](#)

[Replies](#)



**gojimmypi** February 3, 2019 at 10:19 AM

Sorry for the delay in responding.... I believe that since Atmel Studio asks for the location of the Arduino IDE - that the MKR WAN 1300 support comes from the Boards Manager in the Arduino IDE. I'll update this blog to show that.

[Reply](#)

**Anonymous** January 20, 2019 at 5:35 AM

Wow, the format, position and non-population of that SWD connector really are awful aren't they? Nice work, thanks for blogging it.

[Reply](#)



**bentprong** April 18, 2019 at 2:06 PM

Thank you SO much for the time it took you to prepare this tutorial. You literally saved me hours! Good job.

[Reply](#)



**gojimmypi** April 20, 2019 at 8:38 AM

I'm glad it was helpful. Thanks for the feedback. :)

[Reply](#)



**Richard Mansell** July 11, 2019 at 8:03 AM

Thank you so much for this blog post. You are the only one I've found that showed the squid wires connected correctly! No one else ever showed wire #1 (red wire) being connected to the VTG (not even Atmel). I have now successfully programmed my custom SAMD21 board!

[Reply](#)

[Replies](#)



**gojimmypi** July 11, 2019 at 12:26 PM

awesome. I'm glad you found this useful. :)

[Reply](#)




**abaelen** December 24, 2019 at 5:43 AM

Thanks and congrats for this great manual, it was very helpful and instructive. It are these posts that help the opensource community building knowledge. (and arduino MKR1010 needs it...)

[Reply](#) [Delete](#)

Enter your comment...

 Comment as: abaelen (Goog ▾)

Sign out

Publish

Preview

☐ Notify me

comments are welcome, but I prefer not to allow links to promotions or other unrelated services.

Newer Post


Home

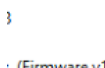
Older Post


Subscribe to: [Post Comments \(Atom\)](#)

ULX3S and IceStudio

Today I learned that IceStudio supports the ULX3S ! I thought I'd first try it in WSL Ubuntu. I'm not sure if X-Windows server is...

- 

**JTAG Debugging for ESP32**  
Notes and information on JTAG Debugging the ESP32 WROOM-32 (aka DevKitC, aka ESP32\_Core\_Board\_V2) I started off my day thinking I'd ta...
- 

**Converting a Blue Pill STM32F103 board to a Black Magic Probe**  
Recently, I learned about the really awesome Black Magic Probe - an interesting JTAG and SWD adapter that essentially has its own, built-in...
- 

**OpenDPS with the DPS5015**  
Once again, I found an awesome project via twitter, this time from Johan Kanflo via this article from hackaday folks: <http://hackaday.co...>