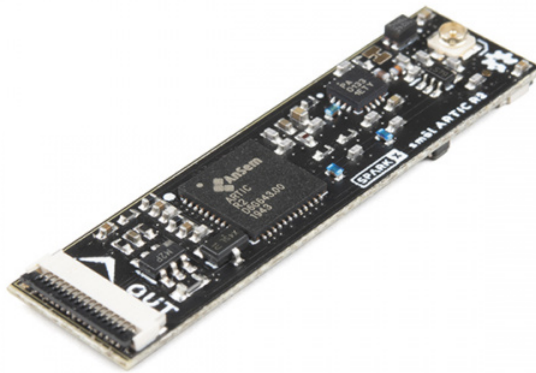


smôl ARTIC R2 Hookup Guide

Introduction

smôl is a new board format and, as the name suggests, they're *really* small!



smôl ARTIC R2

© SPX-18618

The smôl ARTIC R2 Peripheral Board is a complete satellite transceiver for the ARGOS satellite network.

When we designed our ARGOS Satellite Transceiver Shield - ARTIC R2, we knew that people would want to use it for applications like wildlife tracking. In fact, all programs using ARGOS have to be related in some way or other to environmental protection, awareness or study, or to protecting human life. But what if you want to develop something much smaller? Say, a small dart for whale tracking, or a small backpack for avian tracking. Or you just need your battery to last for months. smôl is designed to meet those needs.

Each smôl board measures just 1.60" by 0.42" (40.6mm by 10.7mm). We made the boards just wide enough so we could squeeze USB-C and 16-way Flexible Printed Circuit (FPC) connectors on there. Some of the boards have components on both top and bottom layers which again helps keep the boards small.

smôl boards are designed to stack one on top of the other, using 16-way 0.5mm-pitch FPCs to provide the interconnect from one board to the next. Each board has an **IN** FPC connector on the bottom layer and an **OUT** FPC connector on the top layer. The boards stack in a zig-zag daisy chain; signals and power are passed from

one board to the next up and down the chain through the FPCs.

Required Materials

As a minimum, you're going to need a suitable antenna:



ARGOS Omnidirectional Antenna - 401MHz

● WRL-17523

The ARTIC R2 Peripheral Board is part of the smôl ecosystem. Why not pair it with one of the smôl Processor Boards?



smôl ESP32

● SPX-18619

To be able to reduce the sleep current below 10 μ A, you're going to want to pair the ESP32 with one of our intelligent smôl Power Boards:



smôl Power Board LiPo

● SPX-18622



smôl Power Board AAA

● SPX-18621

Of course, a satellite tracking system needs to know where it is. Why not add a smôl ZOE-M8Q GNSS board to your smôl ecosystem?



smôl ZOE-M8Q

🕒 SPX-18623

Don't forget that you will need Flexible Printed Circuits to connect your smôl boards together. You're going to need one FPC per board. Our 36mm FPC is the perfect length if you want the smôl boards to stack neatly, one on top of the other.



smôl 36mm 16-way Flexible Printed Circuit

🕒 CAB-18731

Need to do some prototyping with smôl? Or want to connect your smôl stack to another Qwiic board? The smôl Header is perfect for that:



smôl Header

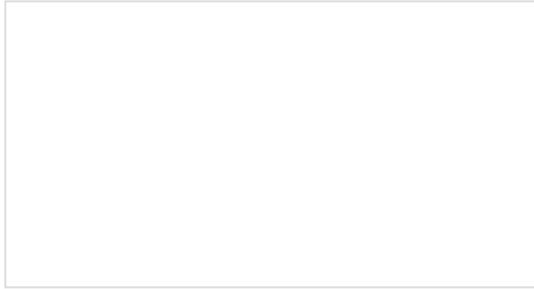
🕒 SPX-18620

Suggested Reading

This is the hookup guide for the smôl ARTIC R2 Peripheral Board. Click the button below if you want to find out more about smôl itself.

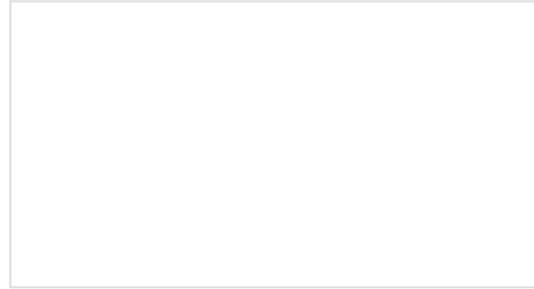
[GET STARTED WITH THE SMÔL HOOKUP GUIDE](#)

We recommend taking a look through the following tutorials if you are not familiar with the concepts covered in them:



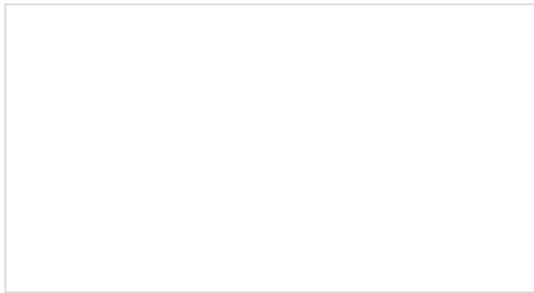
Serial Peripheral Interface (SPI)

SPI is commonly used to connect microcontrollers to peripherals such as sensors, shift registers, and SD cards.



I2C

An introduction to I2C, one of the main embedded communications protocols in use today.



Three Quick Tips About Using U.FL

Quick tips regarding how to connect, protect, and disconnect U.FL connectors.

ARGOS & ARTIC R2

Is your project linked to environmental protection, awareness or study, or to protecting human life? Perhaps you are developing a wildlife tracker, ocean buoy, environmental monitoring system or need to transfer emergency medical information? Do you need to be able to transmit and receive data *anywhere*? If so, this is the smôl product for you! Our smôl ARTIC R2 allows you to send and receive short bursts of data via the ARGOS satellite network, anywhere on Earth including the Polar regions.

The ARGOS satellite system has been around for quite a while. It was created in 1978 by the French Space Agency (CNES), the National Aeronautics and Space Administration (NASA) and the National Oceanic and Atmospheric Administration (NOAA), originally as a scientific tool for collecting and relaying meteorological and oceanographic data around the world. Today, ARGOS is revolutionising satellite communication, adding a constellation of 25 nanosatellites to complement the 7 traditional satellites carrying ARGOS instrumentation. The

first of these, ANGELS, is already in operation and SparkFun were among the first users to transmit data to ANGELS in October 2020. When the constellation is complete, there will be a maximum of 10-15 minutes between satellite passes.

The ARTIC R2 is an integrated, low-power, small-size ARGOS 2/3/4 single chip transceiver. ARTIC implements a message based wireless interface. For satellite uplink communication, ARTIC will encode, modulate and transmit provided user messages. For downlink communication, ARTIC will lock to the downstream, demodulate and decode and extract the satellite messages. The ARTIC can transmit signals in frequency bands around 400MHz and receive signals in the bands around 466MHz, in accordance with the ARGOS satellite system specifications.

The smôl ARTIC R2 has been tested and certified by Kinéis. Compared to other satellite communication systems, the smôl ARTIC R2 has a much lower current draw and will work with a very simple, very lightweight quarter-wave wire antenna. The ARTIC R2 chipset on our board operates from 3.3V and the on-board flash memory enables fast boot times. If you don't need the full transmit power, or want to conserve your battery life, you can transmit at reduced power too thanks to opto-isolated gain control.

Our Arduino Library makes it really easy to get up and running with ARGOS. We've provided a full set of examples which will let you: configure the ARTIC R2 chipset; predict the next satellite pass; receive allcast and individually-coded messages; transmit messages using ARGOS 2, 3 and 4 encoding. There are dedicated examples for the smôl ARTIC R2.

The ARGOS satellite system is restricted to specific programs and applications. Please check that your project meets these requirements before buying hardware. CLS (France) and the Woods Hole Group (America) will be able to advise if your project meets the requirements. *"To meet system use requirements, all programs using Argos have to be related in some way or other to environmental protection, awareness or study, or to protecting human life."*

Hardware Overview

In this section we'll cover what's included on the smôl ARTIC R2 Peripheral Board.

ARTIC R2

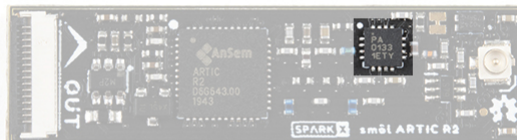
The heart of the smôl ARTIC R2 is, of course, the ARTIC R2 transceiver itself. This is a clever chip containing a Digital Signal Processor (DSP) which modulates transmit messages and demodulates received messages. The DSP can boot from on-board flash memory or from an external microcontroller via SPI. When transmitting, it produces a 1mW (0dBm) output signal which is fed to a separate power amplifier.



Our Arduino Library does all of the heavy lifting for you. By default, the library will tell the ARTIC R2 DSP to boot from the on-board flash memory. However, by changing one line of code, you can instead boot via SPI with your microcontroller providing the firmware for the DSP.

RF Amplifier

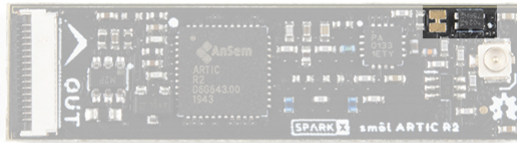
During transmit, the RFPA0133 power amplifier boosts the 0dBm (1mW) signal from the ARTIC R2.



By default, the amplifier uses full gain and boosts the signal to approximately 25.8dBm (380mW). If you are using ARGOS 2 or 3 modulation and are transmitting from a 'noisy' environment, like a city, then you are probably going to need to use full power to ensure your messages get through. However, if you are using ARGOS 4 modulation and/or are transmitting from a 'quiet' environment, like the tundra or the ocean, then you will be able to transmit at reduced power.

Gain Control

There are two ways to reduce the sm&L ARTIC R2 transmit power. You can adjust the gain through software and the on-board opto-isolated gain control circuit.

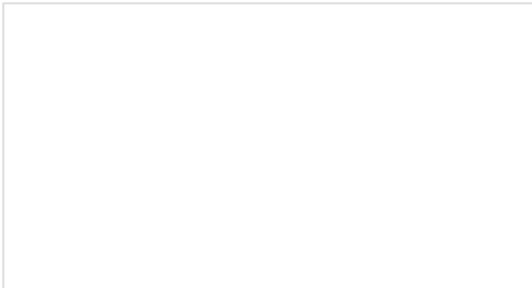


Our Arduino Library can reduce the gain for you. If you call:

```
myARTIC.attenuateTXgain(true);
```

from inside your code, the opto-isolator will pull the RFPA0133's G8 pin low, reducing the gain by approximately 5dB. This also has the advantage of reducing the transmit current by approximately 80mA.

If you are experimenting with ARGOS 4 modulation, you may want to reduce the gain even further. You can do that by opening the split-pad jumper next to the opto-isolator. Opening the jumper will pull the RFPA0133's G16 pin low and will reduce the gain by approximately 11dB. If you haven't used jumpers before, please check out our tutorial.



How to Work with Jumper Pads and PCB Traces

Handling PCB jumper pads and traces is an essential skill. Learn how to cut a PCB trace, add a solder jumper between pads to reroute connections, and repair a trace with the green wire method if a trace is damaged.

Flash Memory

If you turn the smôl ARTIC R2 over, you will be able to see the small flash memory chip.



During production testing at SparkFun, we program the flash memory with the ARTIC R2 firmware (ARTIC006) and a Platform ID allocated by CLS. You will need to register the Platform ID on your ARGOS account to activate it.

GPIO Expander

The larger chip on the bottom of the board is a PCA9536 I²C GPIO expander.

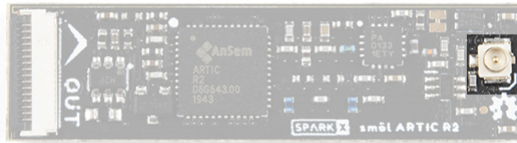


sm01 only supports two GPIO signals, so we added a separate I²C GPIO expander to allow the ARTIC R2 RESETB, INT1 and BOOT signals, plus the RF amplifier G8 pin, to be controlled via I²C.

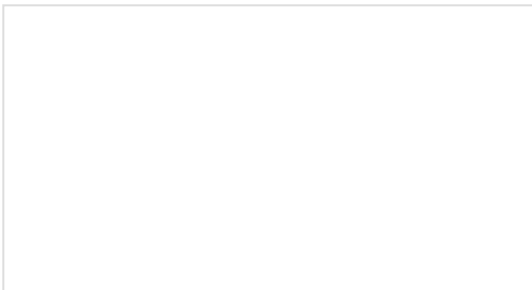
You won't need to communicate with the GPIO expander directly, our Arduino Library will do that for you, we're just letting you know why that chip is there.

Antenna

The antenna connection for the ARTIC R2 is u.FL.



Check out our tutorial if you haven't used u.FL before:



Three Quick Tips About Using U.FL
Quick tips regarding how to connect, protect, and disconnect U.FL connectors.

Warning: Always connect a 401MHz antenna to your smôl ARTIC R2 before connecting the power. Transmitting with no antenna attached will damage the RF power amplifier.

Current Draw

If you are using the smôl ARTIC R2 to develop your own wildlife tracker, you will of course be very interested in how much current the board draws.

The fantastic Oti Arc Power Analyzer has allowed us to capture the ARTIC R2's exact peak transmit current draw at all four gain settings, and to study the average current drawn when the chip is idle. We've used this data to help optimize the code in this example to extend the battery life as much as possible when using the smôl ARTIC R2 with the smôl ESP32 Processor Board, smôl ZOE-M8Q GNSS Board and the smôl Power Board LiPo.

Maximum Gain (G16 High, G8 High)

The picture below shows the current draw captured by Otii Arc for a complete transmit cycle using this example. The gain here is set to maximum.



Having a hard time seeing? Click the image for a closer look.

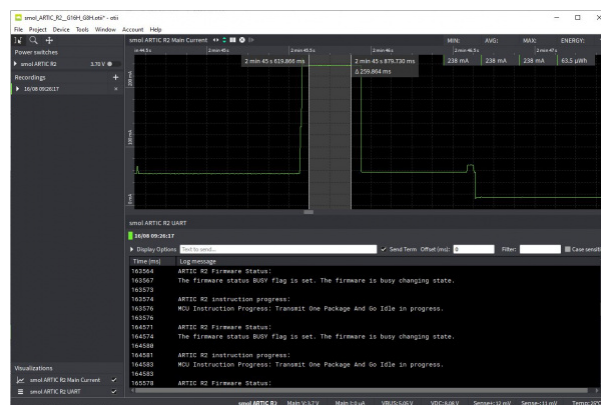
At the start of the transmit cycle, the ESP32 Processor Board is woken up by the Power Board. It turns on the ZOE-M8Q GNSS board and waits for a fix. The current draw here is approximately 100mA for ~10 seconds.

Once the ESP32 has a location fix, it can calculate the time of the next ARGOS satellite pass. The power board woke the ESP32 up 1 minute before the start of the next pass, so the ESP32 goes in and out of light sleep for the next ~30 seconds.

The ARTIC R2 is powered on at the 40 second mark and the first of five transmits takes place just after the one minute mark.

The code example transmits five times on each satellite pass. The gap between the transmits is 90 seconds with a mandatory $\pm 10\%$ jitter. In between, you can see the ESP32 going in and out of light sleep with the ARTIC R2 being powered up 20-30 seconds before the next transmit is due.

The phenomenal data captured by Otii Arc allows us to zoom right in on the transmit itself and capture the true maximum current draw:



Having a hard time seeing? Click the image for a closer look.

It's then child's play to extract the data we need!

Current Draw Summary

Below is a summary of the peak transmit current draw for the four ARTIC R2 gain settings. The RFPA0133 G8 pin is controlled by software. The G16 pin is controlled by the split-pad jumper. The sm01 stack was powered by a 3.7V LiPo battery.

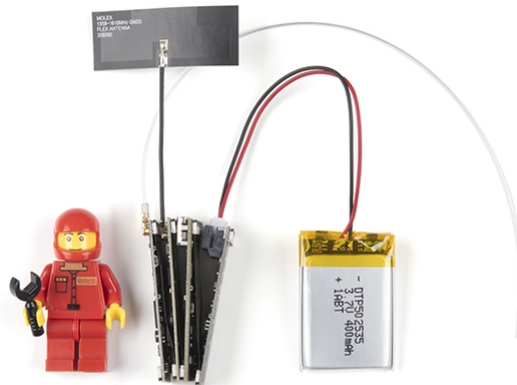
Please note: the Peak TX Current in this table is the total peak current drawn by the ESP32 + ARTIC R2 + Power Amplifier during the actual transmit

G16	G8	Approx. TX Power (dBm)	Peak TX Current (mA)
High	High	26	238
High	Low	21	190
Low	High	15	133
Low	Low	4	114

Otii Arc lets us collect the other data we need:

- LiPo Battery Voltage: 3.7V
- Average Current Draw during a complete 5*TX transmit cycle (maximum gain): 29.9mA
- Average Current Draw during a complete 5*TX transmit cycle (minimum gain): 24.9mA
- Average continuous ESP32 + ZOE-M8Q GNSS Current Draw: 97mA
- Current Draw during deep sleep: 6.4µA
- Power Consumption during a complete 5*TX transmit cycle (maximum gain): 13.3mWh
- Power Consumption during a complete 5*TX transmit cycle (minimum gain): 11.1mWh

We can then use this data to predict the battery life based on the transmit power and the number of transmissions per day. The satellite pass-prediction code in our Arduino Library can calculate the timing of the satellite passes with the highest elevation each day to maximize the chance of the transmission being received. Transmitting five times per day at maximum power, we could expect a 2000mAh battery to last approximately 530 days! Even the modest 400mAh battery pictured below should last for more than 100 days.



smôl Specifics

Interfaces:

- SPI and I²C
 - SPI Chip Select: CS0 (via waterfalloing)
 - PCA9536 GPIO Expander I²C Address: 0x41

GPIO:

- Uses GPIO0 for power control (via waterfalling)
- Leave GPIO0 floating or pull low to disable power for the ARTIC R2
- Pull GPIO0 high to enable power for the ARTIC R2
- Power is controlled automatically via the SparkFun ARTIC R2 Arduino Library

Arduino Example: Satellite Detection

If you are using the sm⁰l ESP32 Processor Board, you are going to want to install the CP210x USB Driver and Arduino Boards Package for the ESP32 first. Please see the sm⁰l ESP32 Hookup Guide for more details.

The sm⁰l ARTIC R2 peripheral board is fully compatible with the SparkFun ARGOS ARTIC R2 Arduino Library. You can install the library through the Arduino IDE Library Manager by searching for **SparkFun ARGOS ARTIC**. Alternatively, you can grab the library from GitHub or can download it as a zip file by clicking the button below:

SPARKFUN ARGOS ARTIC R2 ARDUINO LIBRARY (ZIP)

The library contains tried-and-tested dedicated examples for the sm⁰l ARTIC R2. The code below is a stripped-down version of Example3_SatelliteDetection. Copy and paste the code into a new window in the Arduino IDE and upload to the ESP32. The ARTIC R2 will try to detect a satellite for up to 10 minutes. You may wish to log into the ARGOS website and predict when the next satellite pass will take place before running this example.

```

#include <Wire.h> //Needed for I2C to ARTIC R2 GPIO
#include <SPI.h>
#include "SparkFun_ARGOS_ARTIC_R2_Arduino_Library.h" // http://librarymanager/All#SparkFun_ARGOS
_ARTIC_R2
ARTIC_R2 myARTIC;

int CS_Pin = 5;           // sm01 CS0 = ESP32 Pin 5
int ARTIC_PWR_EN_Pin = 27; // sm01 GPIO0 = ESP32 Pin 27

void setup()
{
  Serial.begin(115200);
  Serial.println(F("ARGOS sm01 ARTIC R2 Example"));

  Serial.println(F("ARTIC R2 is booting..."));

  Wire.begin(); // Needed to communicate with the I2C GPIO chip on the sm01 ARTIC R2
  SPI.begin();

  //myARTIC.enableDebugging(); // Uncomment this line to enable debug messages on Serial

  // Begin the ARTIC: enable power and boot from flash
  if (myARTIC.beginSmol(CS_Pin, ARTIC_PWR_EN_Pin) == false)
  {
    Serial.println("ARTIC R2 not detected. Please check the sm01 stack-up and flexible circuits.
Freezing...");
    while (1)
      ; // Do nothing more
  }

  Serial.println(F("ARTIC R2 boot was successful.));

  myARTIC.setTCXOControl(1.8, true); // Set the TCXO voltage to 1.8V and autoDisable to 1

  myARTIC.setSatelliteDetectionTimeout(600); // Set the satellite detection timeout to 600 seconds

  Serial.println(F("Starting satellite detection...));

  // Start satellite detection
  // The ARTIC will start looking for a satellite for the specified amount of time.
  myARTIC.sendMCUinstruction(INST_SATELLITE_DETECTION);
}

void loop()
{
  delay(1000);

  // Read the ARTIC R2 status register
  ARTIC_R2_Firmware_Status status;
  myARTIC.readStatusRegister(&status);

  // Check the interrupt 2 flag. This will go high if satellite detection times out

```

```
if (status.STATUS_REGISTER_BITS.DSP2MCU_INT2)
{
  Serial.println(F("INT2 pin is high. Satellite detection has timed out!"));
}
// Check the interrupt 1 flag. This will go high when a satellite is detected
else if (status.STATUS_REGISTER_BITS.DSP2MCU_INT1)
{
  Serial.println(F("INT1 pin is high. Satellite detected!"));
}

// Check the instruction progress
// checkMCUinstructionProgress will return true if the instruction is complete
ARTIC_R2_MCU_Instruction_Progress progress;
boolean instructionComplete = myARTIC.checkMCUinstructionProgress(&progress);

if (instructionComplete)
{
  Serial.println(F("Satellite detection is complete! Freezing..."));
  while (1)
    ; // Do nothing more
}
}
```

Troubleshooting

Not working as expected and need help? SparkX products are rapidly produced to bring you the most cutting edge technology as it becomes available. These products are tested but come with no guarantees. Live technical support is not available for SparkX products. Head on over to our forum for support or to ask a question and we will get back to you as soon as we can.

Resources and Going Further

For more information about the smôl ARTIC R2, check out the following links:

smôl ARTIC R2 Documentation:

- Schematic
- Eagle Files
- ARGOS Chipset Info Sheet
- ARTIC R2 User Datasheet v1.1
- GitHub Hardware Repo
- Arduino Examples
- SparkFun ARGOS ARTIC R2 Arduino Library

smôl Documentation:

- smôl Hookup Guide

Thanks! The ARTIC R2 circuit is a remix of the reference design kindly provided by the Arribada Initiative and Icoteq Ltd.

