Skitter Critter Wiring Notes

This Skitter robot uses the Huzzah32 feather ESP32 controller connected to a Crickit Featherwing board to increase the I/O count. The Featherwing connects to the Critter board that maps pins back to the Skitter main board via the MXP 34 pin connector.



Figure 1. Huzzah32 Feather - "The Brains"





Figure 2. Crickit Featherwing - "The Expander"



Figure 4. Skitter Main Board - "The Muscle"

Figure 3. Skitter Critter - "The Mediator"

Motor Connections

We use the motor control port of the Crickit Featherwing board as I/O pins only. The (black) Skitter main board inside the chassis does the actual motor control.



5V Supply Connection

We use the 5V supply from the Crickit Featherwing board to power the 5V side of the Level Shifter on our Critter board. The level shifters allow the 5V sensors to safely work with the 3.3V Huzzah32 and Crickit. The Sig. line is tied to LED control on the Skitter main board (MXP connector pin 26).



3V Supply Connection

We use the 3V supply from the <u>Huzzah32 featherwing</u> to power the 3V side of the Level Shifter on our Critter board. Connect via jumper (lone red wire) as shown.



Ultrasonic Connection

The Critter board has Echo and Trig. pins for the ultrasonic sensor. The connection to the controller is somewhat arbitrary. For the demo code we chose pin 17 for the Echo pin and pin 21 for the Trigger pin. These connection pass through a level shifter.



Line Follower Encoder Connections

These connections go through level shifters. Due to pin limitations on the Skitter main board, the line follower switch and S5 don't have connections to the Skitter main board. To use these, separate jumpers from those pins on the line follower would need to be connected to the Switch and S5 pins on the Critter board. The rest of the I/O pass to the Skitter main board.



Programming

When programming make sure you plug the USB cable into the Huzzah32 feather port (Marked with green check mark in photo). Due to the layout there are 2 usb ports next to each other. The other is the Crickit Seesaw port. NOTE: It won't hurt anything if you mistakenly plug into the wrong port - you'll just get an error from your programming tool.



Programming Continued:

I have assembled a program (<u>SkitterWebButtonRobotMouseUpDown</u>) that is a good starting point for the Skitterbot. It connects to your network allowing the bot to serve up a control button page. Mouse clicks (up and down) control bot motion. If the ultrasonic sensor detects a "close" object it fires an interrupt which triggers the red led to come on. I tried to add sensor measurement feedback to the page but I don't have the javascript expertise or time to get it working in "real-time". It displays once correctly when the page loads. Like I said, it is a starting point.

Make sure you have the correct board and settings in your IDE. In the Arduino IDE for example it would look like this:

SkitterWebButtonRobotMouseUpDown | Arduino 1.8.13

File Edit Sketch Tools Help

	Auto Format Archive Sketch	Ctrl+T
SkitterWebButt	Fix Encoding & Reload	
/*******	Manage Libraries	Ctrl+Shift+I
Rui Santo:	Serial Monitor	Ctrl+Shift+M
Complete p	Serial Plotter	Ctrl+Shift+L
********/		
//CSK 08/25,	WIFI101 / WIFININA Firmware Updater	
//https://ri	Board: "Adafruit ESP32 Feather"	>
//Intterupts	Upload Speed: "921600"	>
// Load Wi-I	Flash Frequency: "80MHz"	>
<pre>#include <w:< pre=""></w:<></pre>	Partition Scheme: "Default"	>
#include <a< th=""><th>Core Debug Level: "Info"</th><th>></th></a<>	Core Debug Level: "Info"	>
#include <s< th=""><td>Port</td><td>></td></s<>	Port	>
<pre>#include <ne< pre=""></ne<></pre>	Get Board Info	
#define ECH(#define TRI(Programmer Burn Bootloader	>