AI.FRAME

WILD FLOWER USER GUIDE V3.0

Cautions!

- Read this quick start guide completely before wiring and applying power to the controller! Errors in wiring can damage the Wild Flower motion capture controller sensors, STM32 or EEPROM Chip, and any peripherals.
- Never reverse the battery power coming in to the controller. Make sure the black wire goes to (-) GND, and the red wire goes to (+) VCC.
- To operate Wild Flower motion capture controller, coding skill and Arduino basics are required.
- Power switch is on the left handle, remember to turn off the controller when it is not using.
- The controller will not start when battery is low. Battery charging port is a 3-pin port under the back cover, charge the battery in time, or will damage the battery.

Wild Flower motion capture controller

The Wild Flower motion capture controller is a motion capture controller allows you to

control your robots, drones, video games and any execution devices with your body motions by wearing it on your shoulder. It has 8 high-precision angle sensors on the two arms, a 9 buttons pad on the left handle and a 3 degrees joystick on the right handle. The design of the Wild Flower   motion capture controller help it to follow the human body movements effectively. Since the Wild Flower motion capture controller is powered by a STM32 chip and base on an Arduino compatible board Maple Rev 3, coding skill and Arduino basics are needed.
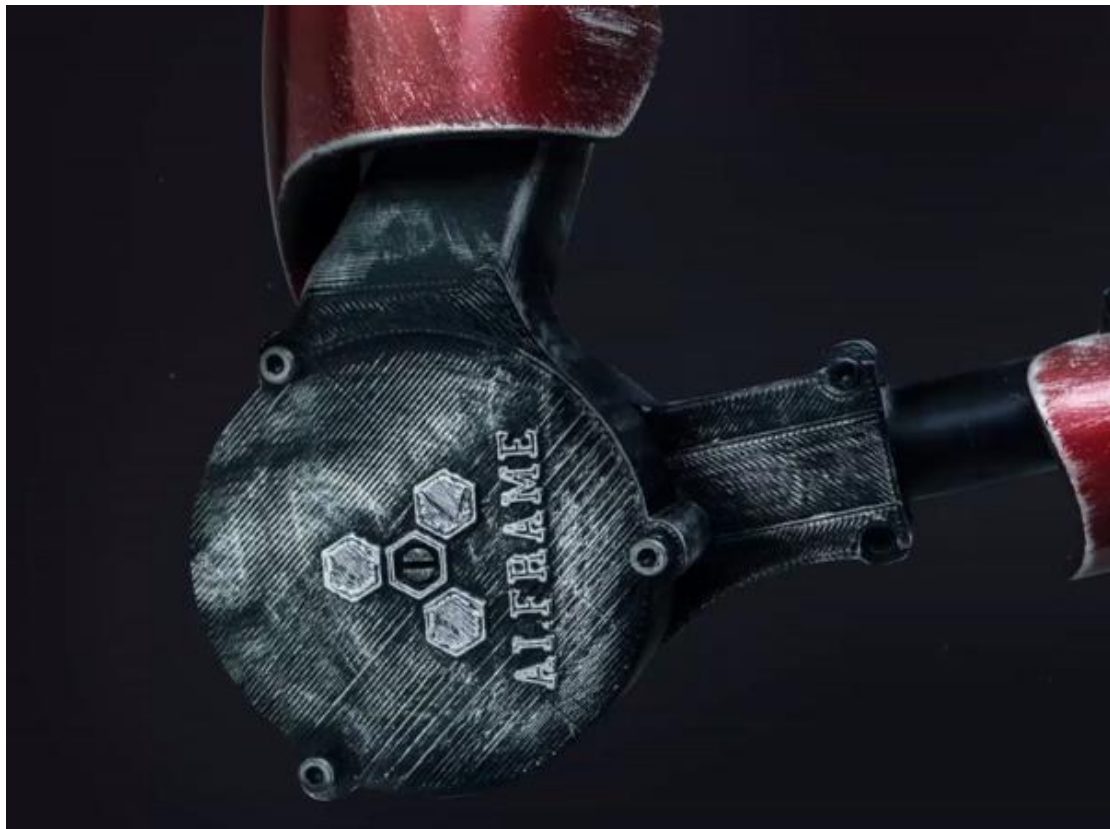
Maple quick start guide:

http://docs.leaflabs.com/docs.leaflabs.com/index.html
http://leaflabs.com/

Feathers:

- STM32 based hardware
- Program via USB interface with PC
- Compatible with Arduino coding
- The precision of angle signal is 0.088°
- Communicating speed up to 1M bps
- UART and USB port supported

Angle Sensor Module

The angle sensor modules used on the Wild Flower motion capture controller is designed by Ai.Frame team. They are all connected together via 485 interface, with an unique ID number, the position of each module are shown in the picture below.
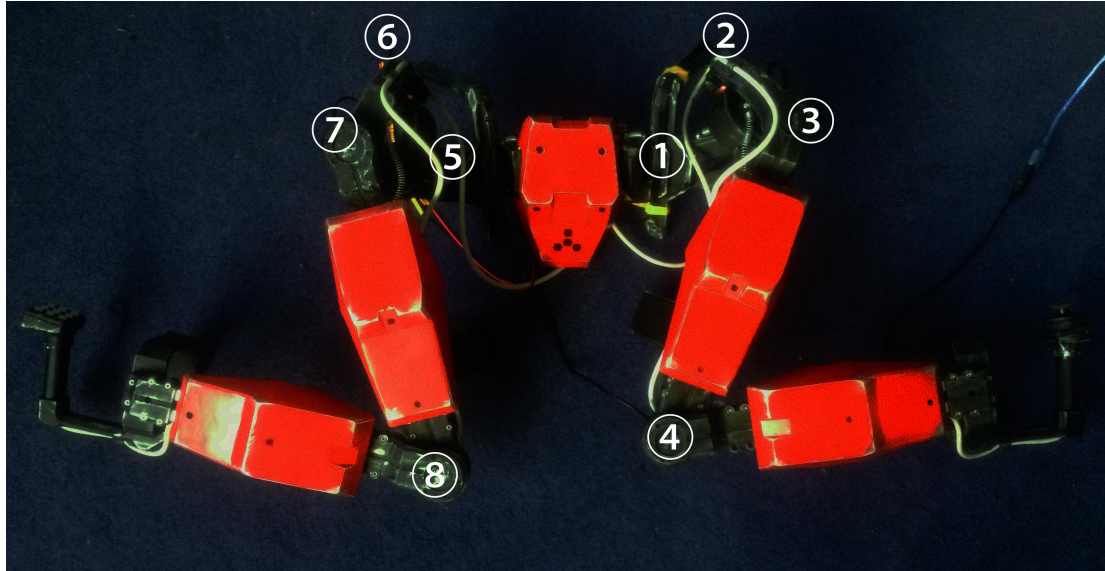
Command Formatting Of Angle Sensor Module:

An effective command for the sensor modules contain 6 parts, each of which is represented by hexadecimal value, shown in the figure below:

Command Head	Module ID	Command Length	Command code	Data	Check Byte
FFFF (2 Bytes)	ID (1 Byte)	LEN (1 Byte)	CODE (1 Byte)	DATA(1)... DATA(N) (0~254 Bytes)	CHECK (1 Byte)

FF FF: Command start flag.
ID: ID of target module.
LEN: Command length. Equals to the length of the data plus 2, in byte. Command length = Data length + 2.
DATA: Data ready to be sent.
CHECK: Check byte. CHECK = ~(ID + CODE + DATA(1) + DATA(2) + ... + DATA(N))%0xFF.
RETURN: An effective return command also contain 6 parts data. A full return command will be like this: FF FF ID LEN CODE DATA1 DATA2 ... DATAN CHECK.
CODE:

AI.FRAME

No	Command Name	Command Code	Function
1	CMD_STATUS_GET	0x1	Get the status of the sensor module.
2	CMD_ID_SET	0x2	Set the modules ID. 1 byte entry parameter data needed. Example: FF FF 00 03 02 01 F9. This example sets the module ID 0 to ID 1. The default ID is 0.
3	CMD_BAUD_SET	0x3	Set the module baud rate. 1 byte entry parameter data needed. Example: FF FF 00 03 03 01 F8. This example sets the module 0 baud to 1, means 500K bps. The entry parameters and their corresponding baud rate: 0: 1M 1: 500K 2:400K 3: 256K 4: 200K 5: 115200 6: 57600 7: 19200 8: 9600. The default baud rate is 1M.
4	CMD_BAUD_GET	0x4	Get the module baud rate. No entry parameter data needed. Example: FF FF 00 02 04 F9. This example gets the baud rate of the module 0, it will return 1 byte data, which contain the message of the baud rate.
5	CMD_DIR_GET	0x5	Forbidden.
6	CMD_DIR_SET	0x6	Forbidden.

No	Command Name	Command Code	Function
7	CMD_ANGLE_GET	0x7	Get the angle data of the module. No entry parameter data needed. Example: FF FF 01 02 07 F5. This example gets the angle data of the module 1, it will return 2 byte data containing the angle output. The first byte data means the low 8 bit of the angle output, the second byte means the high 8 bit. The angle output range is from 0 to 4096.
8	CMD_RAWANGLE_GET	0x8	Forbidden.
9	CMD_ANGLE_RESET	0x9	Restore module factory settings.

Buttons Pad

The Wild Flower motion capture controller has a 9- button pad on its left handle, The

buttons functions can be customized to any usages. Such as, controller the lights on a robot, enabling gun trigger for shooting games...

These buttons are connected directly to the maple pin shown in the figure below.

Button NO.	1	2	3	4	5	6	7	8	9
Maple Pin NO.	22	23	2	3	12	11	19	27	28

Joystick

The Wild Flower motion capture controller also has a 3- axis joystick on its right handle, the outputs of axis X and Y are the analog signal, and the output of the axis Z is digital signal.
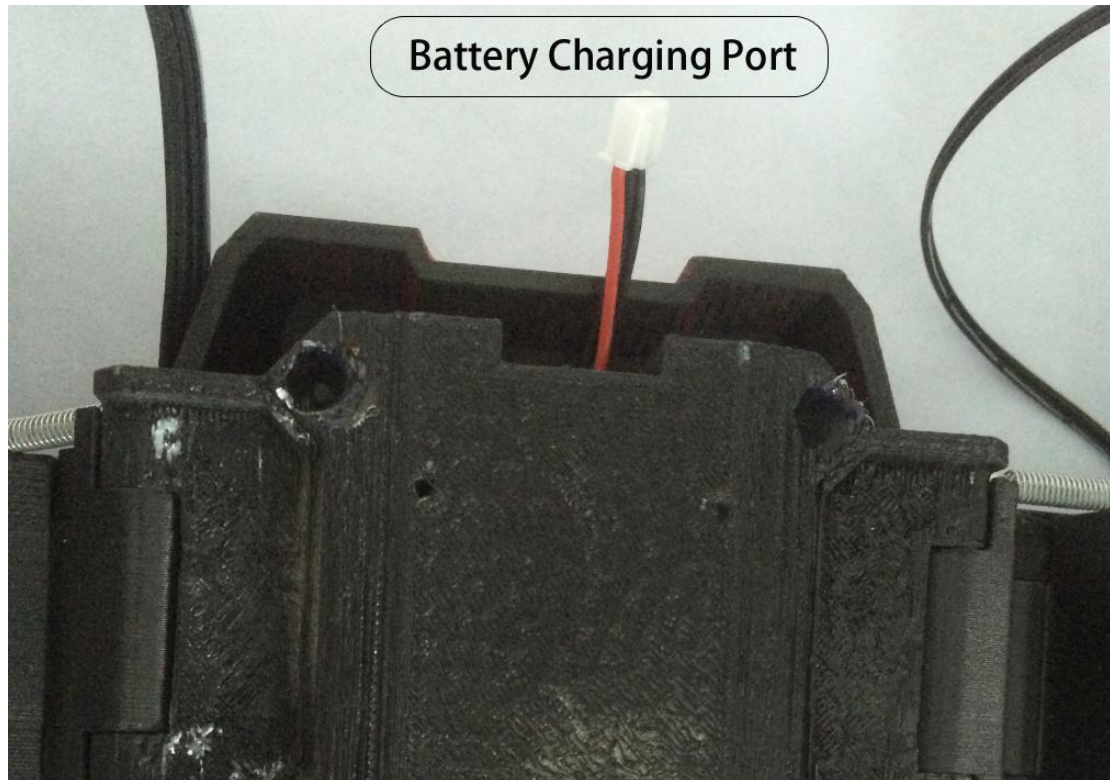
These axes are connected directly to the maple pin shown in the figure below.

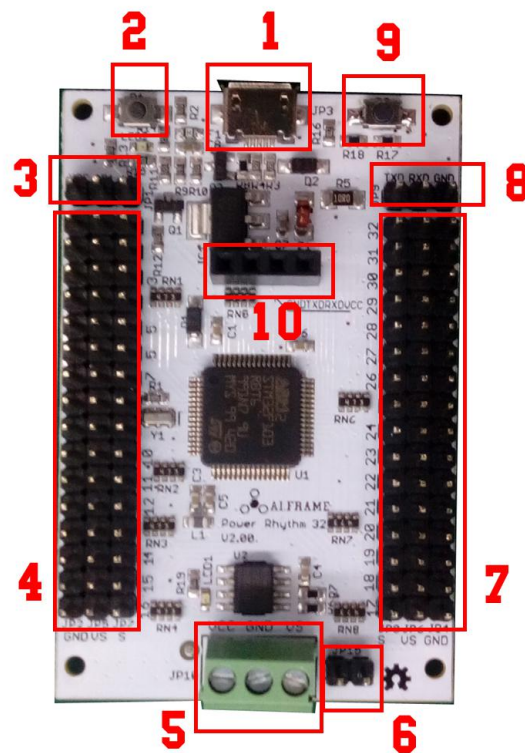| Joystick Axis NO.	X	Y	Z
Maple Pin NO.	16	15	21

Battery

The Wild Flower motion capture controller uses a 7.4V Li-Po rechargeable battery, it must be charged when battery is low, or the battery will be damaged.
The charging port is a 3-pin port under the back cover.

Battery Charging Port

Install The Drivers:

- Open the back cover, you will see a Maple controller:

1. Micro USB port
2. Reset button P1
3. Analog input AD0, AD1, and 3.3V DC output VCC
4. Signals output 1 - 16. Signal 8 and signal 9 is the RX and TX pin of UART2.
5. Power supply for chips (VCC) and sensors (VS)
6. JP15: connect these two pins will connect VCC to VS
7. Signals output 17 - 32.
8. UART1 port
9. Button P2
10. UART1 port

- Download the ZIP file for 32-bits Windows (http://static.leaflabs.com/pub/leaflabs/maple-ide/maple-ide-0.0.12-windowsxp32.zip)
- Extract all the files in the ZIP file to a suitable location on your system (like your Desktop folder).
- First, install DFU drivers (for uploading code to your Maple) using the following steps.
  1.Plug your Maple into the USB port.
  2.Hit the reset button P1 on your Maple (it's the small button at the top left, labeled P1). Notice that it blinks quickly 6 times, then blinks slowly a few more times.
  3.Hit reset again, and this time push and hold the other button during the 6 fast blinks (the button is on the top right; it is labeled P2). You can release it once the slow blinks start.
  4.Your Maple is now in perpetual bootloader mode. This should give you a chance to install the DFU drivers.
  5.Windows should now prompt you for some drivers. In the top level directory of the Maple IDE, point Windows to drivers/mapleDrv/dfu/.
- Next, install serial drivers (for communicating with your Maple using serial over USB).
  1.Reset your Maple and allow it to exit the bootloader (wait for the slow blinking to stop). The Maple will next start running whatever program was uploaded to it last. (New Maples will start running the test program we upload to them before shipping them to you).
  2.Once Maple is running some user code, Windows should prompt you for more drivers. Point windows to driver/mapleDrv/serial. You can now run the Servo_Rhythm_Controller.exe for 32-bits Windows.

Programming

The Wild Flower motion capture controller offers two ways of data transmission, UART

and USB, The UART port is on the right shoulder of the controller with a 3.3V output, it looks like this:

This is the UART2 port on Maple, it is only used for data transmission.
The USB port is for data transmission and firmware uploading. After you install the serial driver it will become a virtual serial com port on PC. Please check the Maple quick start guide for more details.

Examples

There are 3 firmwares for testing your Wild Flower motion capture controller, open ..\Wild_Flower_V3.0\example file you will get 3 files, Buttons, Joint and Joystick, they are used to test the buttons, angle sensors and joystick on the Wild Flower, you will need a COM port monitor to check data coming out from the UART and USB. Please see the coding for more details.

Hardware design: https://github.com/AiFrame/Power_Rhythm_32/tree/master/Eagle_files
Software design: https://github.com/AiFrame/Wild_Flower/
Website: http://aiframe.me
E-mail: 721462533@qq.com, 474995355@qq.com.