Prosthetic Arm How-to

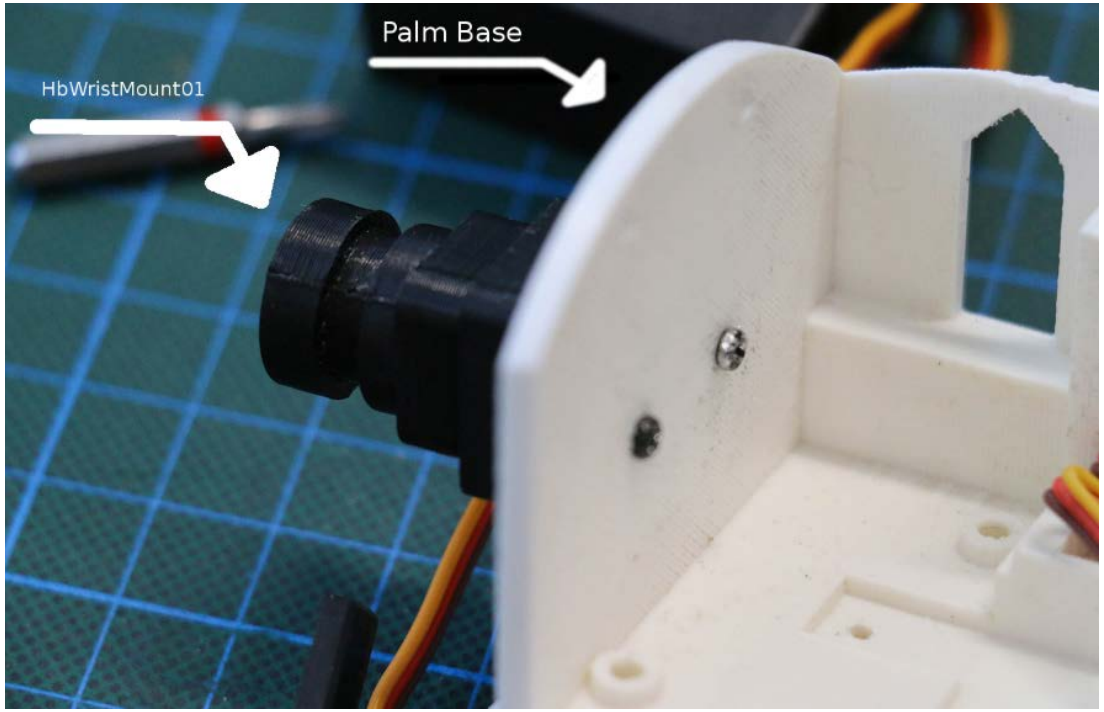
Alvaro Araujo

July, 2018

Berkeley
Wireless Research Center

Mount -wrist

- First of all, screw the wirst on the palm base because once the elements are assembled on the hand (fingers, components), it becomes impossible to do it.



Mount - Middle finger assembly set

- Adjust the holes that will receive the pins with 1.8mm drill (up to 1.9mm)
- Insert axes
- Put together the components as shown from left to right, reading from top to bottom
- Tighten all the components with screws

Mount - Index assembly instruction set

- Enlarge holes with 1.8mm drill (up to 1.9mm max)
- Insert the pin with the spring
- Assemble elements step by step

Mount - Installation of 4 fingers on the palm

- Adjust axes of the palm with 1.8mm drill, adjust to 1.9 max if the axe does not fit
- Place little & ring finger with 2 axes for each fingers
- Adjust, insert & screw **HbShaftStopperB04** to secure the 2 fingers
- Place the middle finger with his 2 axes
- Place index finger with his 2 axes
- Insert and screw **HbShaftStopperA04**

Mount - Installation of 4 fingers on the palm

Mount - Thumb assembly instruction set

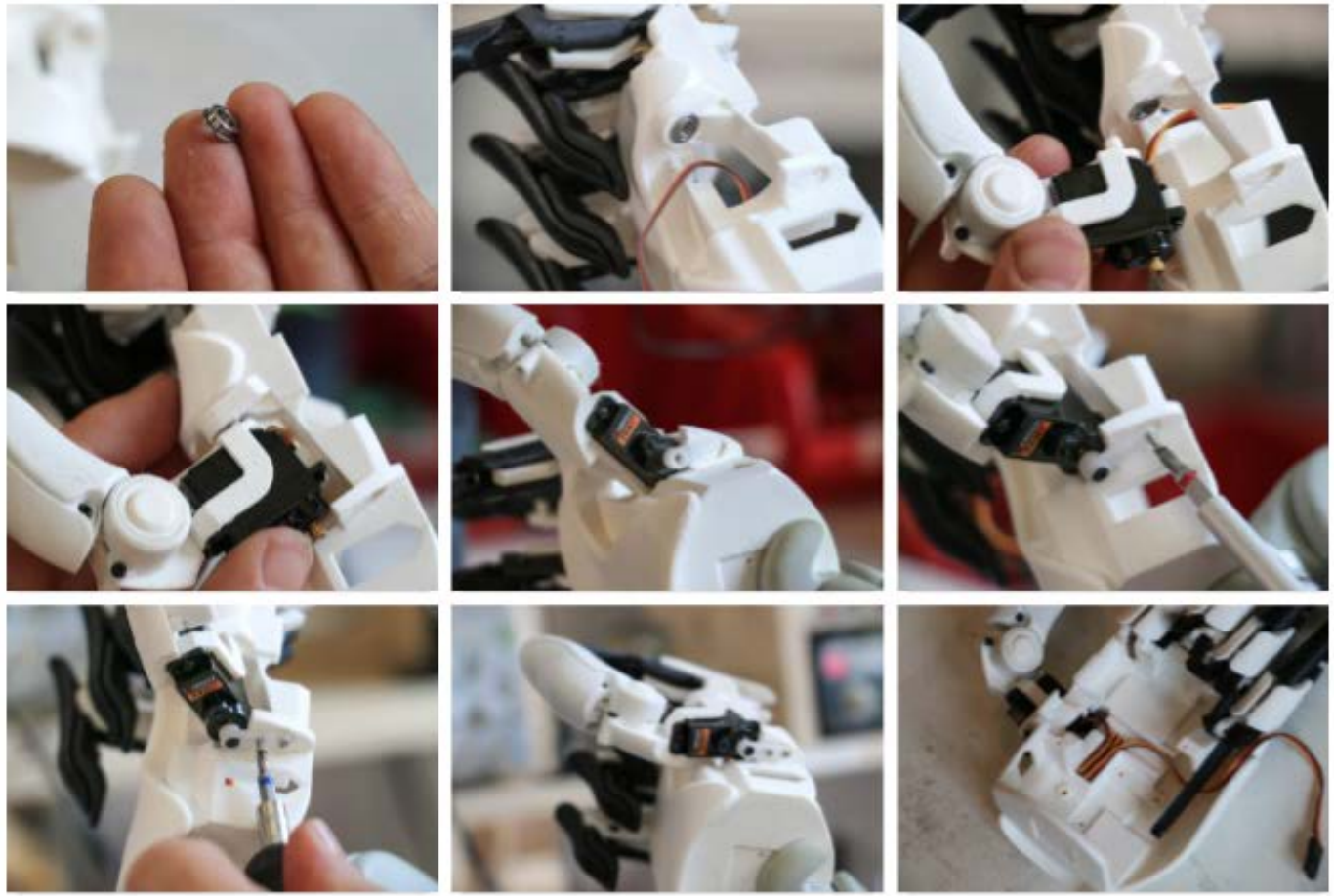
- Check all components and prepare material (3 screws + spring)
- Follow the steps indicated on the pictures, left to right, top to bottom
- Assemble the elements with the ScrewM2L10
- Screw the servo ES08MD on the thumb

Mount - Thumb assembly instruction set

Mount - Set the thumb on the palm

- Insert the mini ball-bearing in the palm as indicated on picture 2
- Place and screw 1 servo ES08MAII in the thumb
- Insert servo's axis in the mini ball-bearing
- Follow the instructions as indicated using 2 screws
- Make sure to fold the cable as indicated, the index motor will be placed above

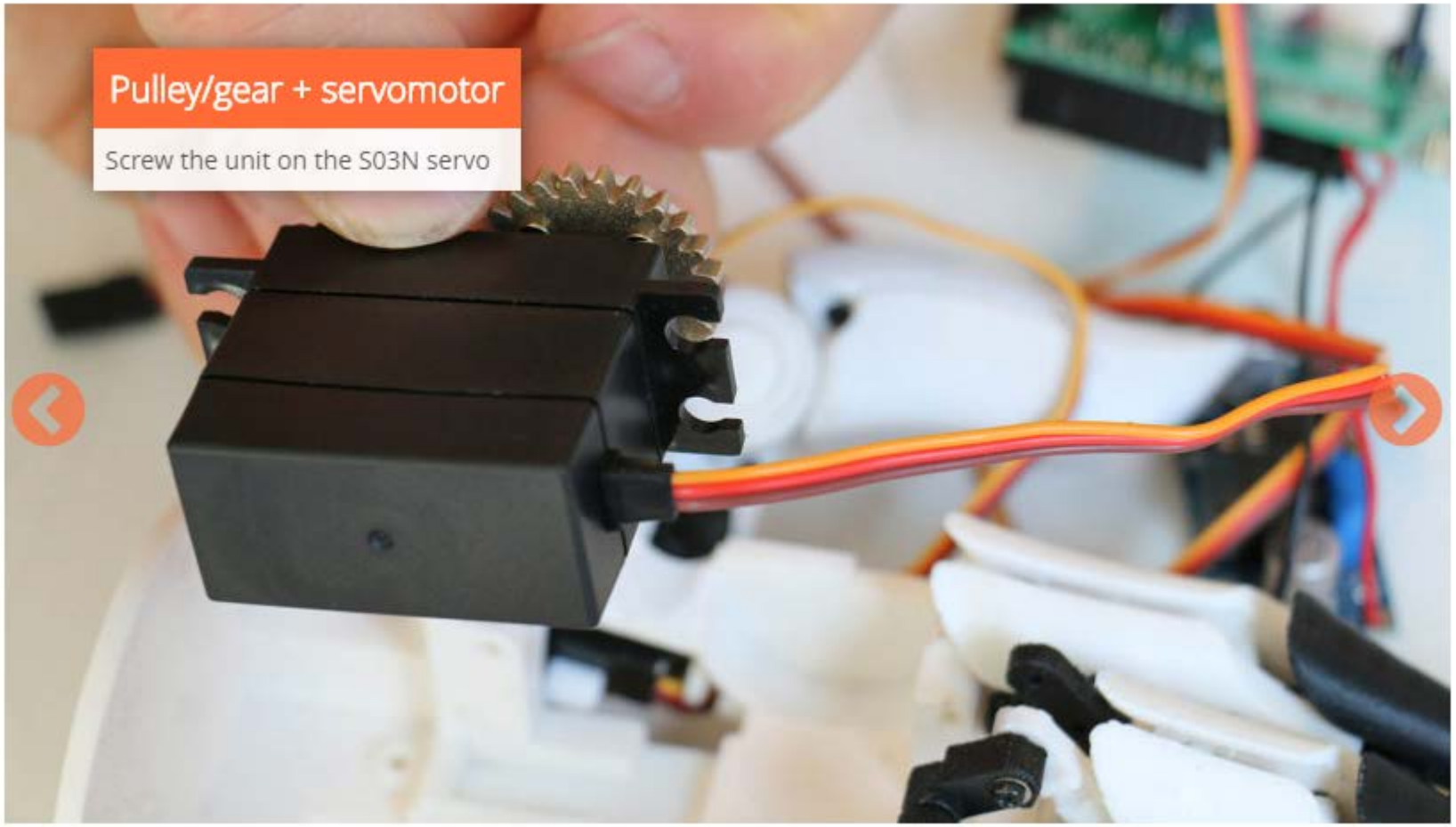
Mount - Set the thumb on the palm



Mount - Index's motor assembly set up

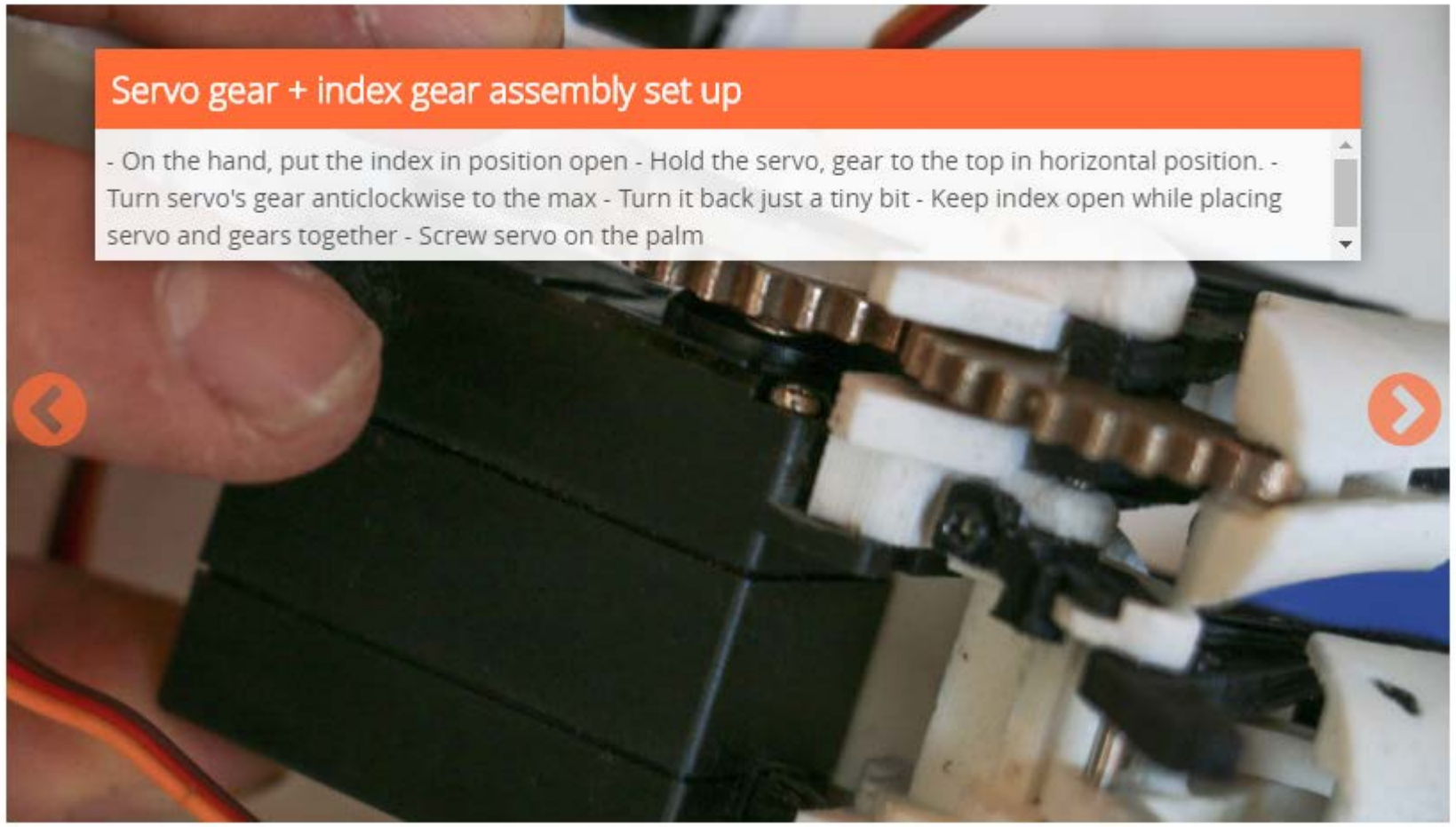
Mount - Index's motor assembly set up

Mount - Index's motor assembly set up



Mount - Index's motor assembly set up

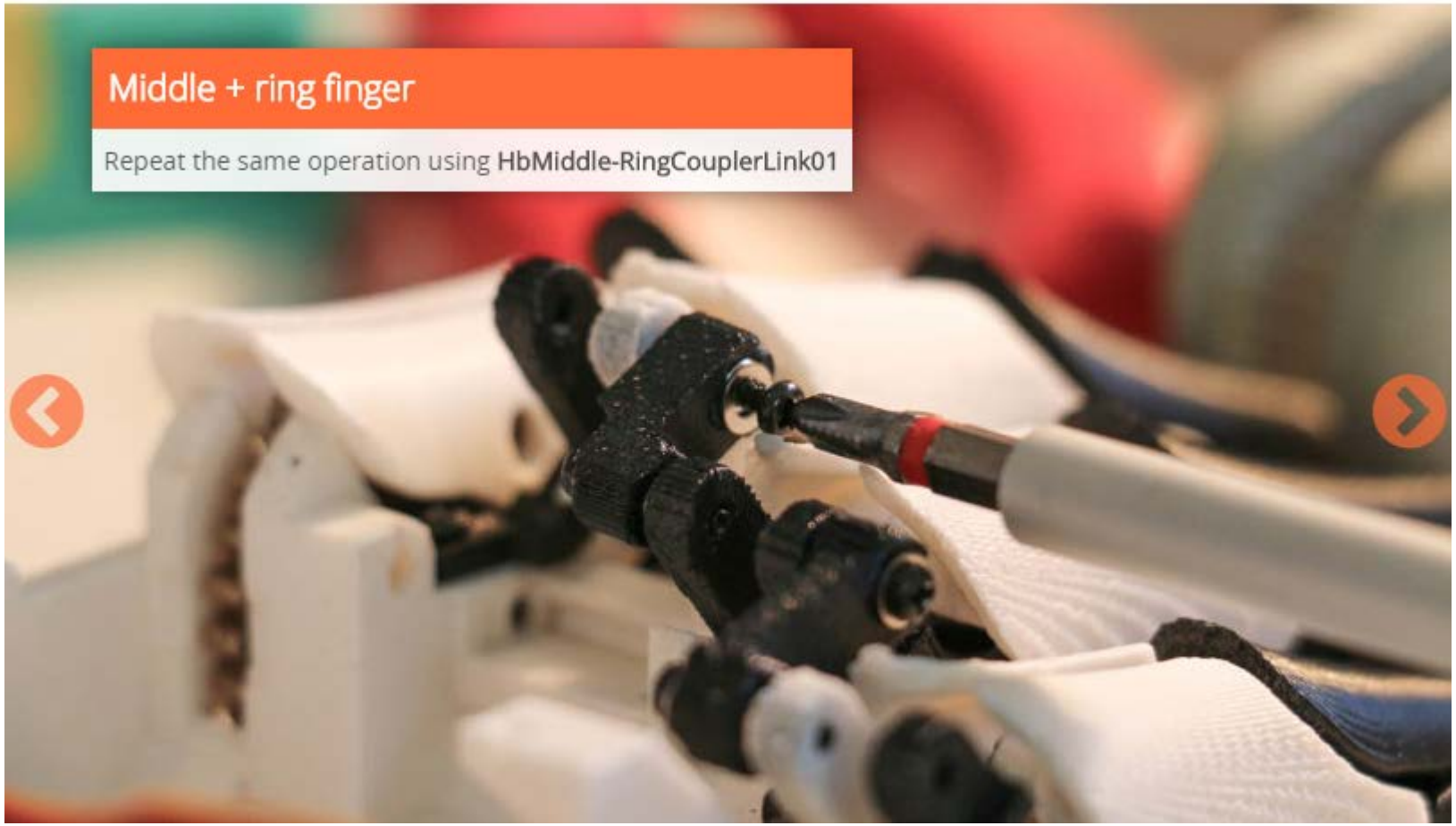
Mount - Index's motor assembly set up



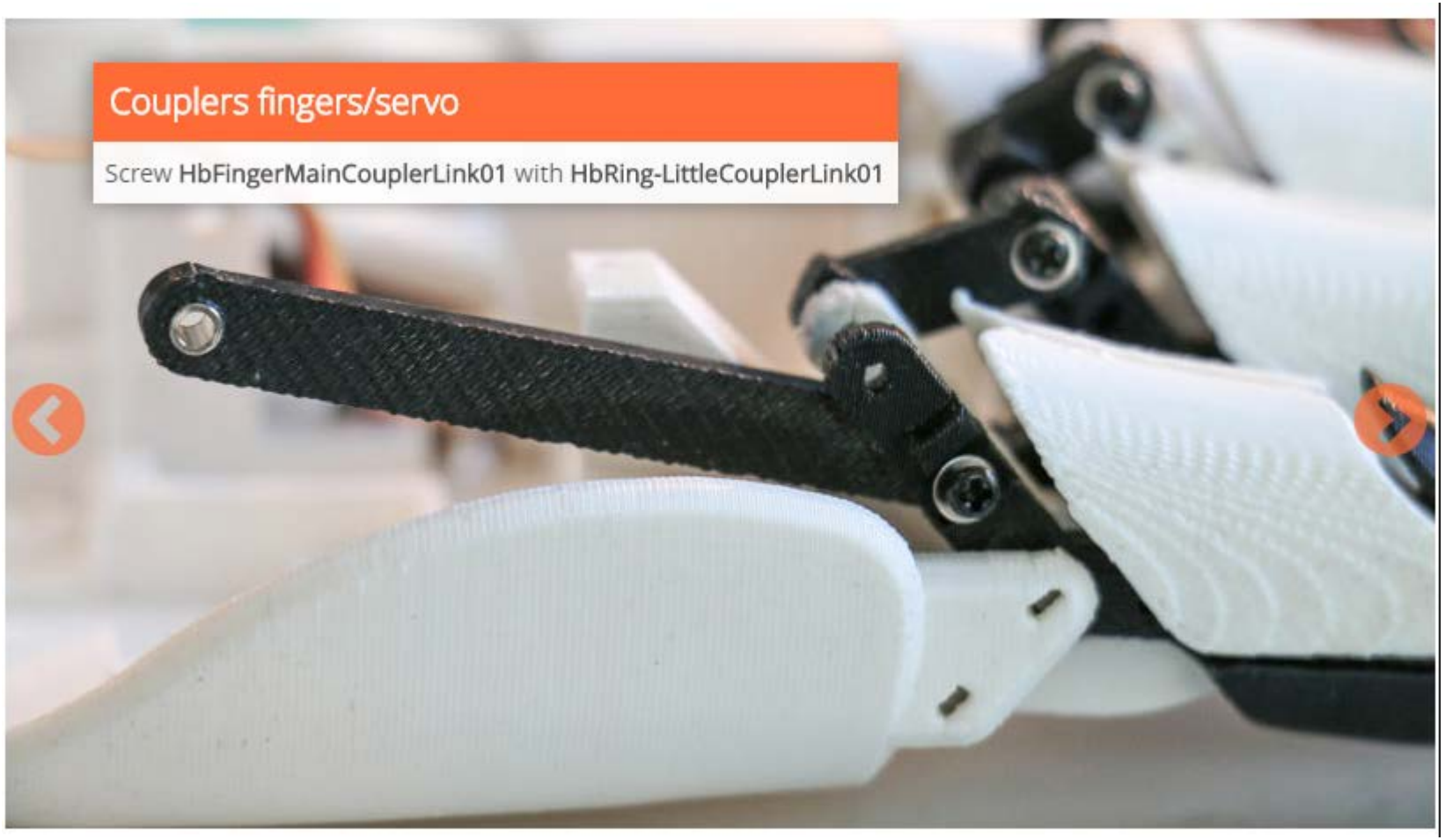
Mount - Coupling 3 fingers + servomotor

Mount - Coupling 3 fingers + servomotor

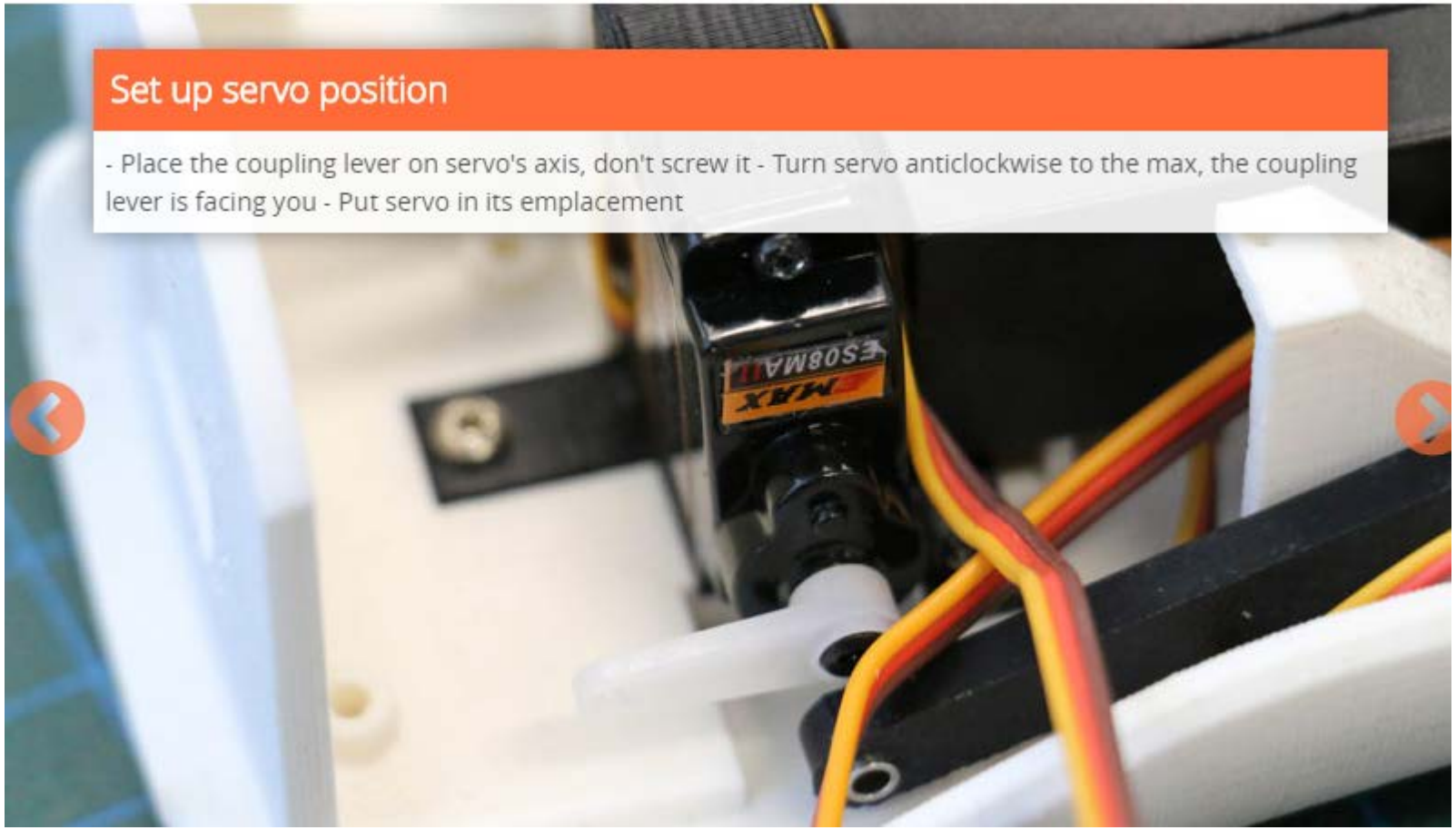
Mount - Coupling 3 fingers + servomotor



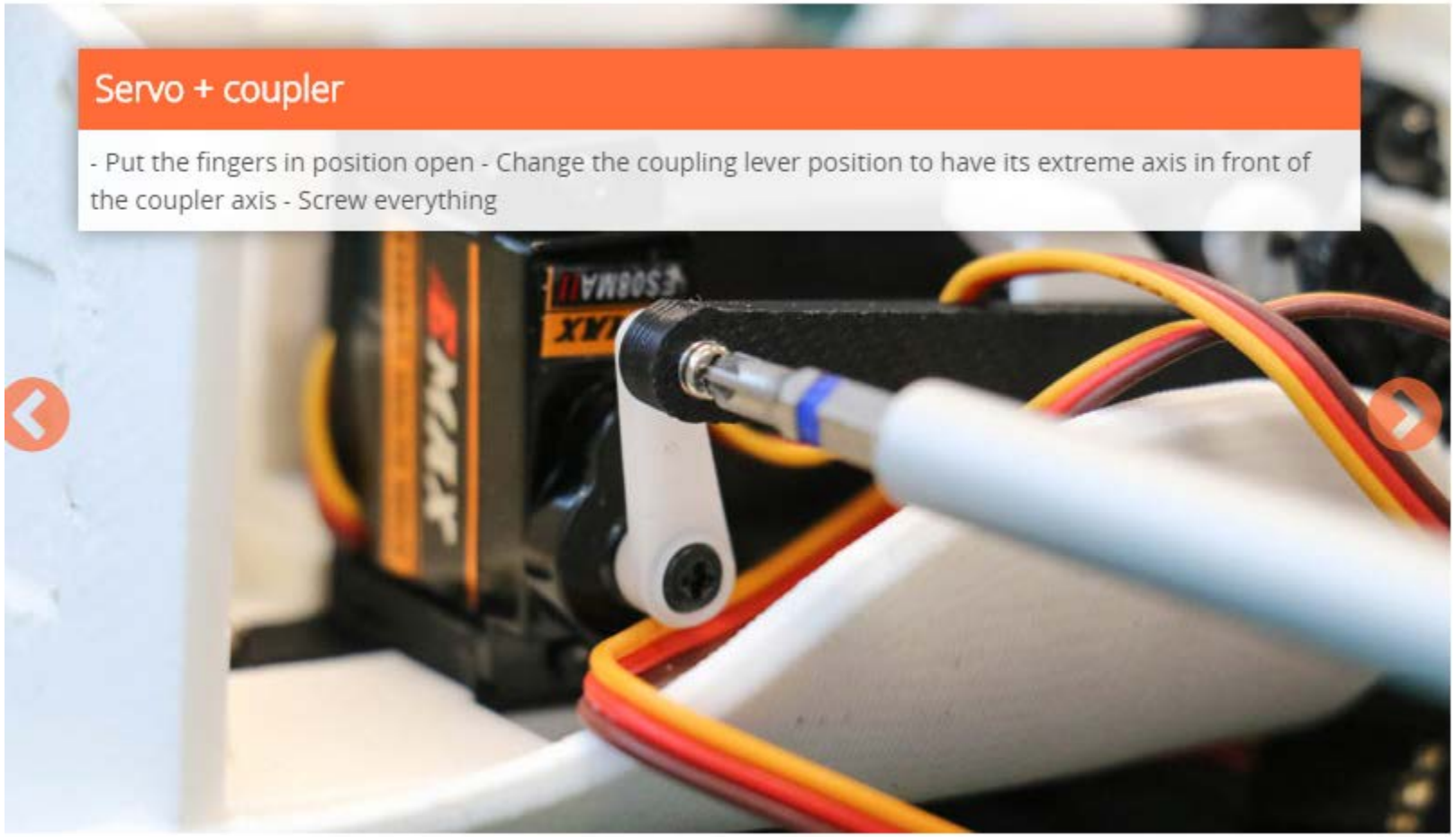
Mount - Coupling 3 fingers + servomotor



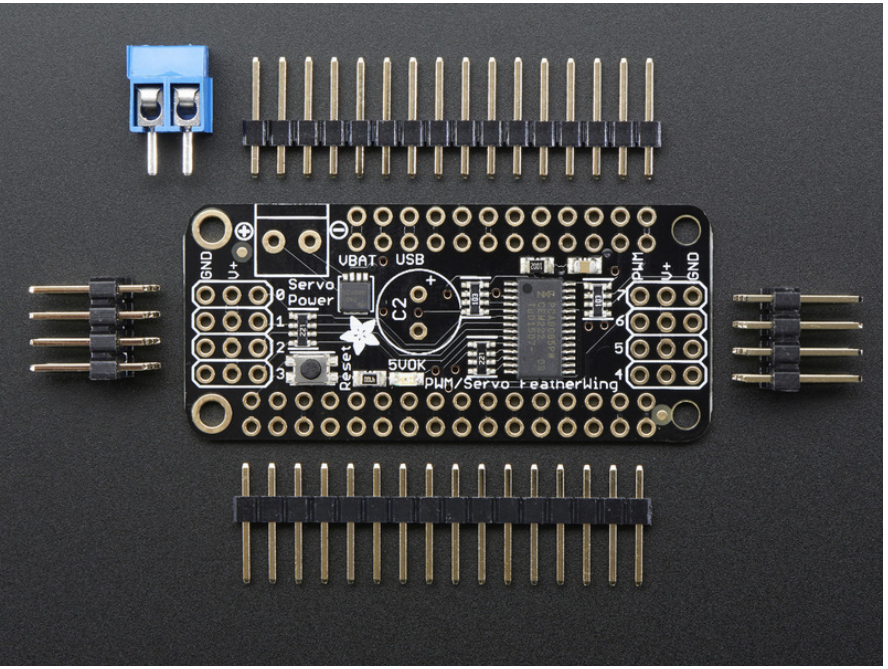
Mount - Coupling 3 fingers + servomotor

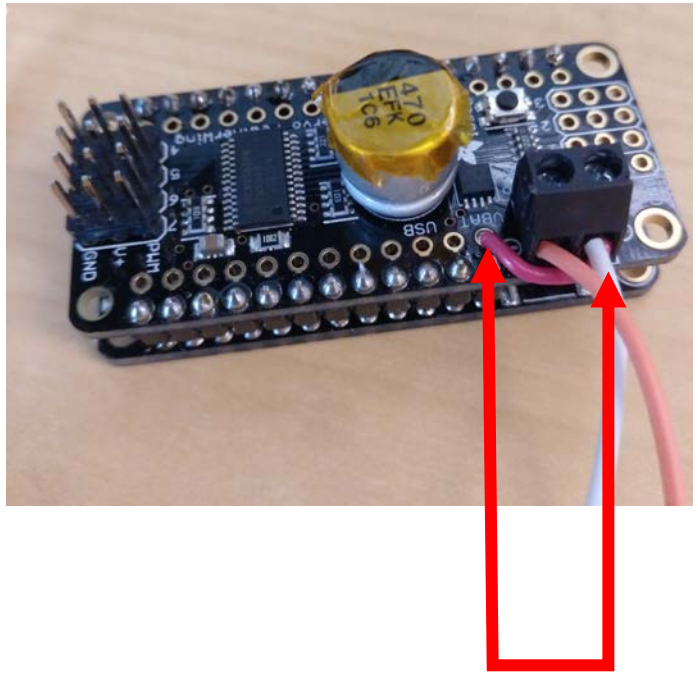


Mount - Coupling 3 fingers + servomotor

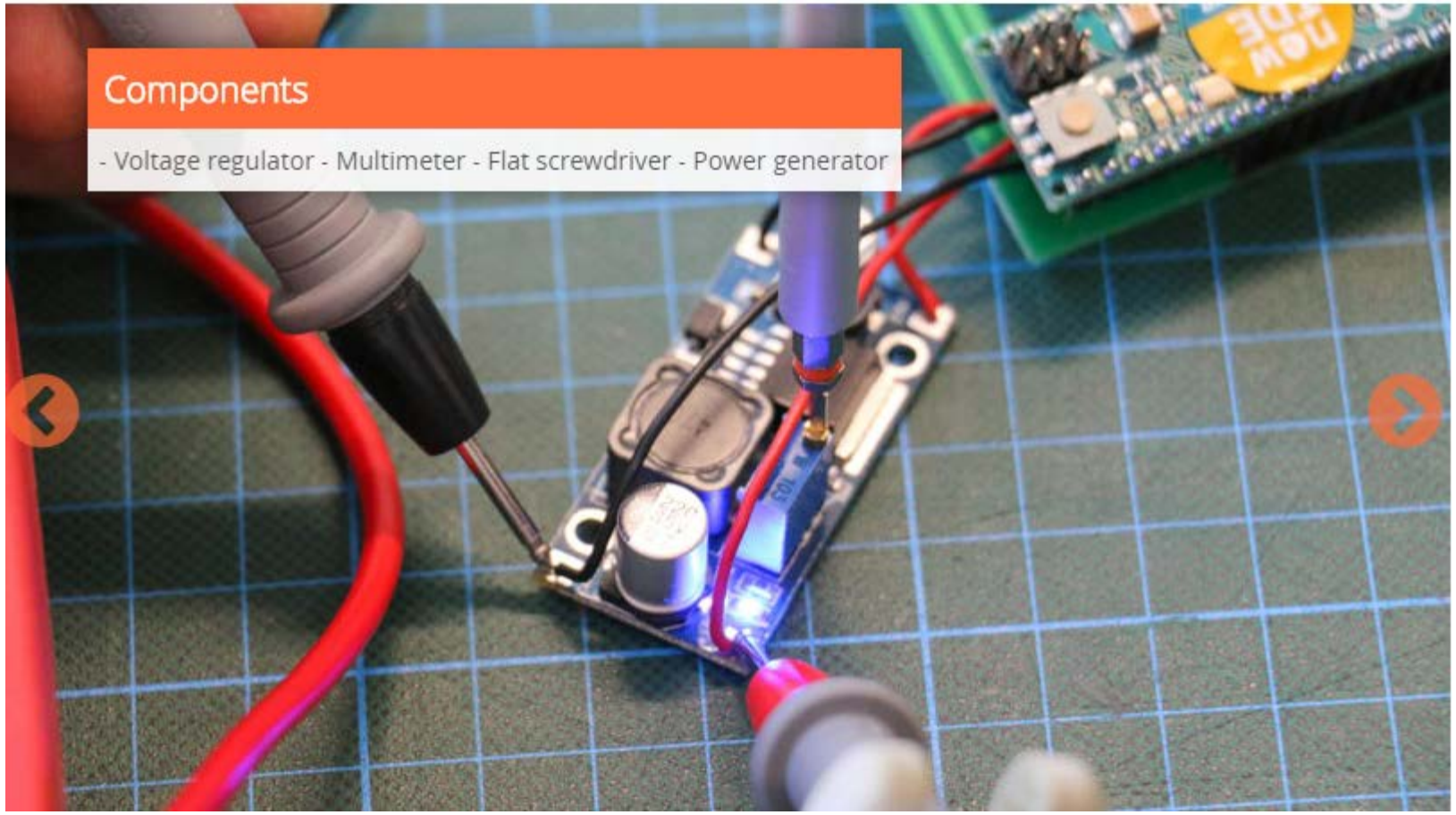


Mount – Soldering electronics

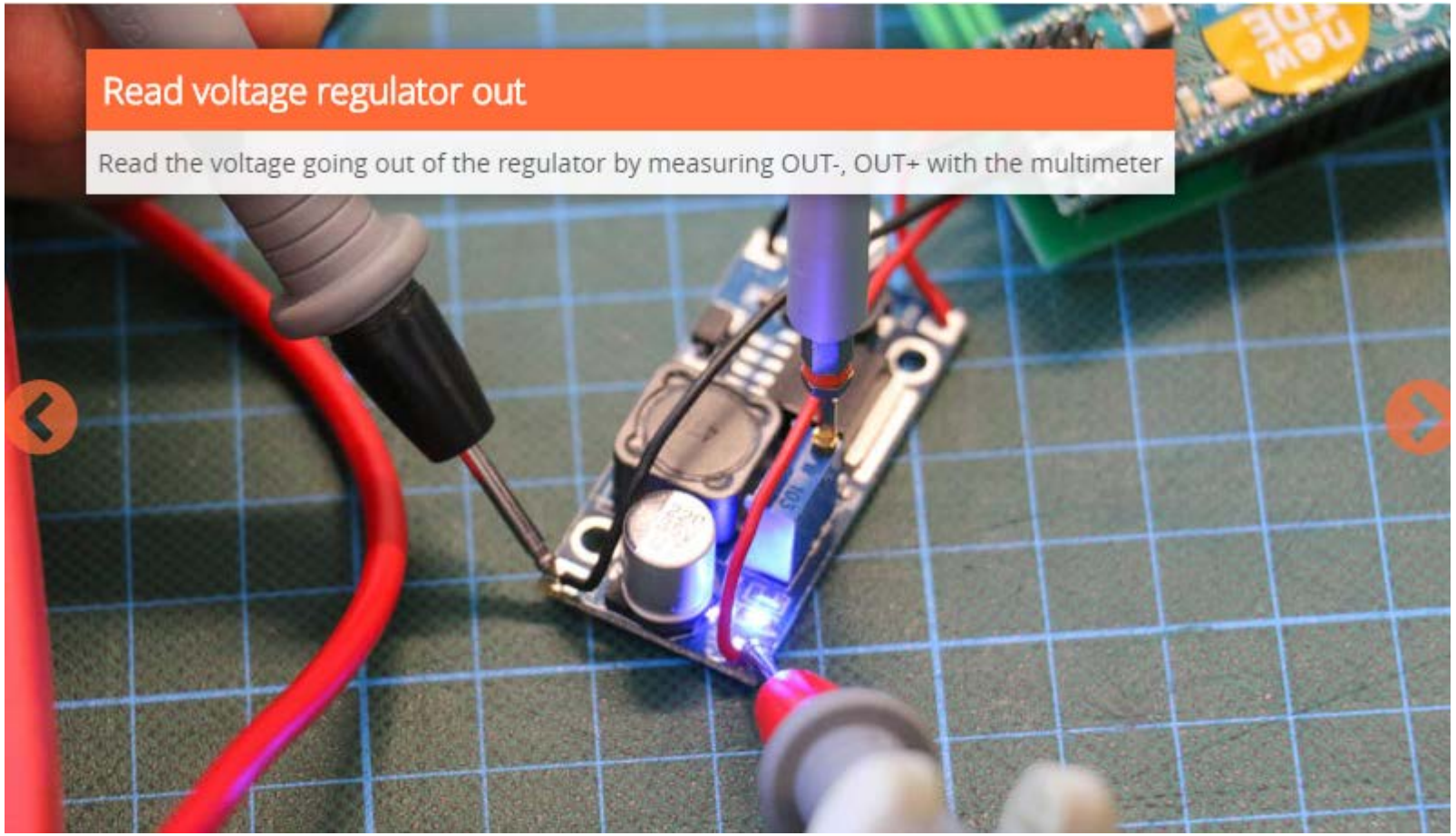




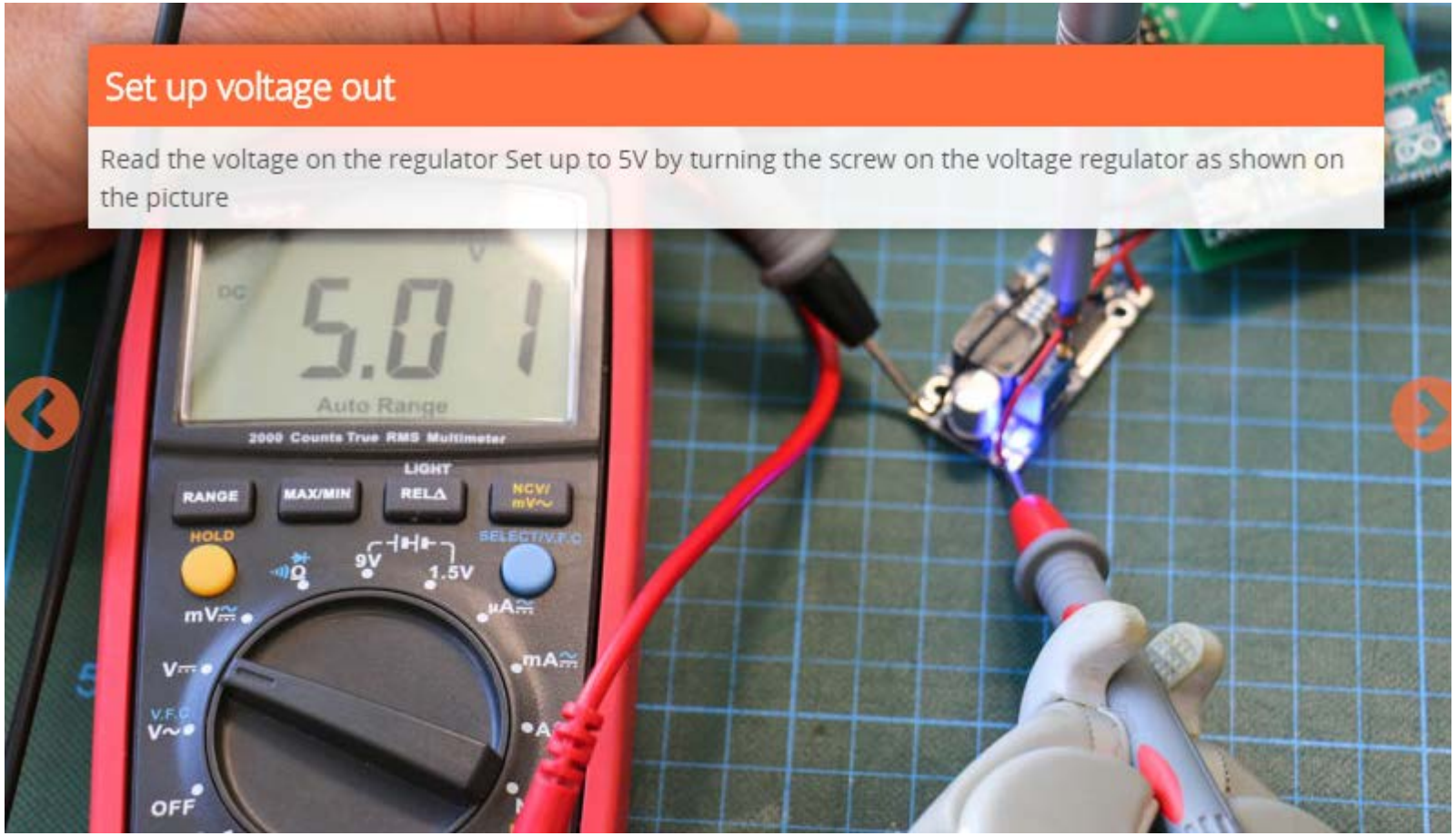
Mount – Voltage regulator adjustment



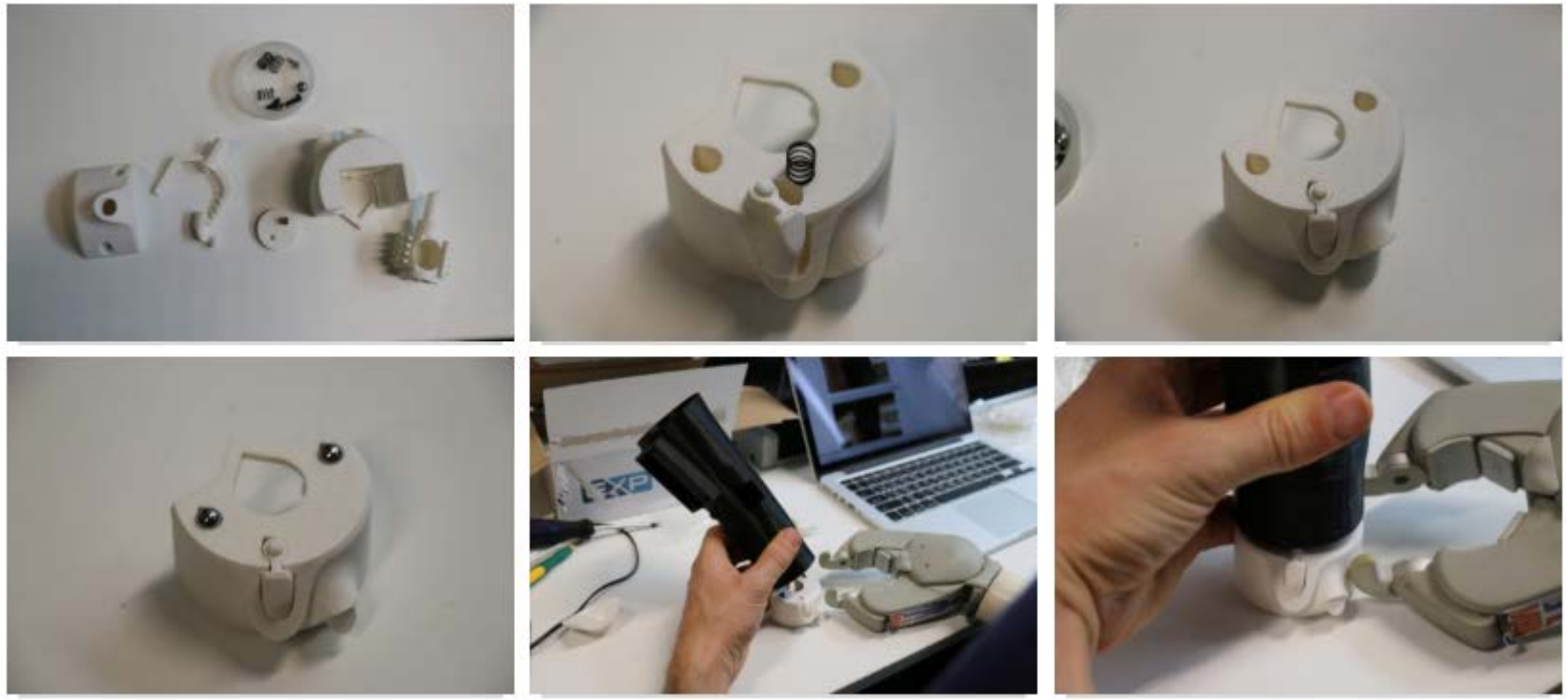
Mount – Voltage regulator adjustment



Mount – Voltage regulator adjustment



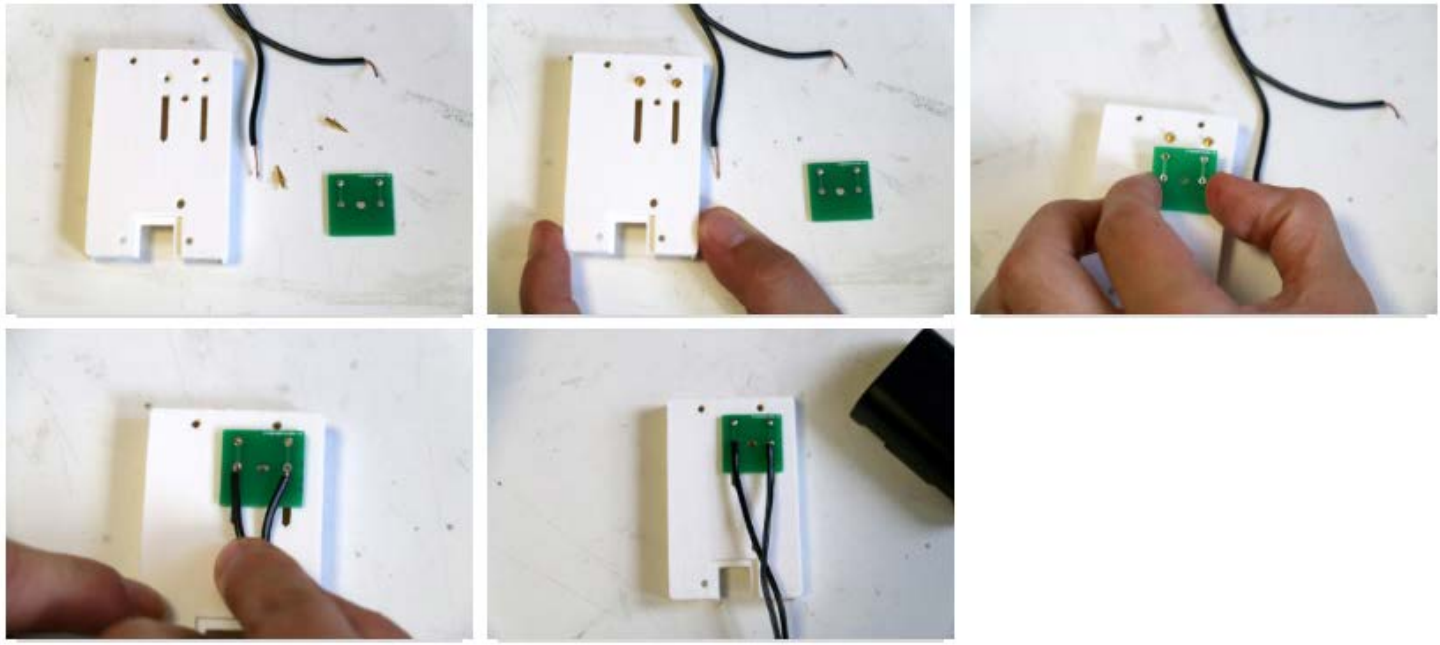
Mount – Wirst & socket assembly set up



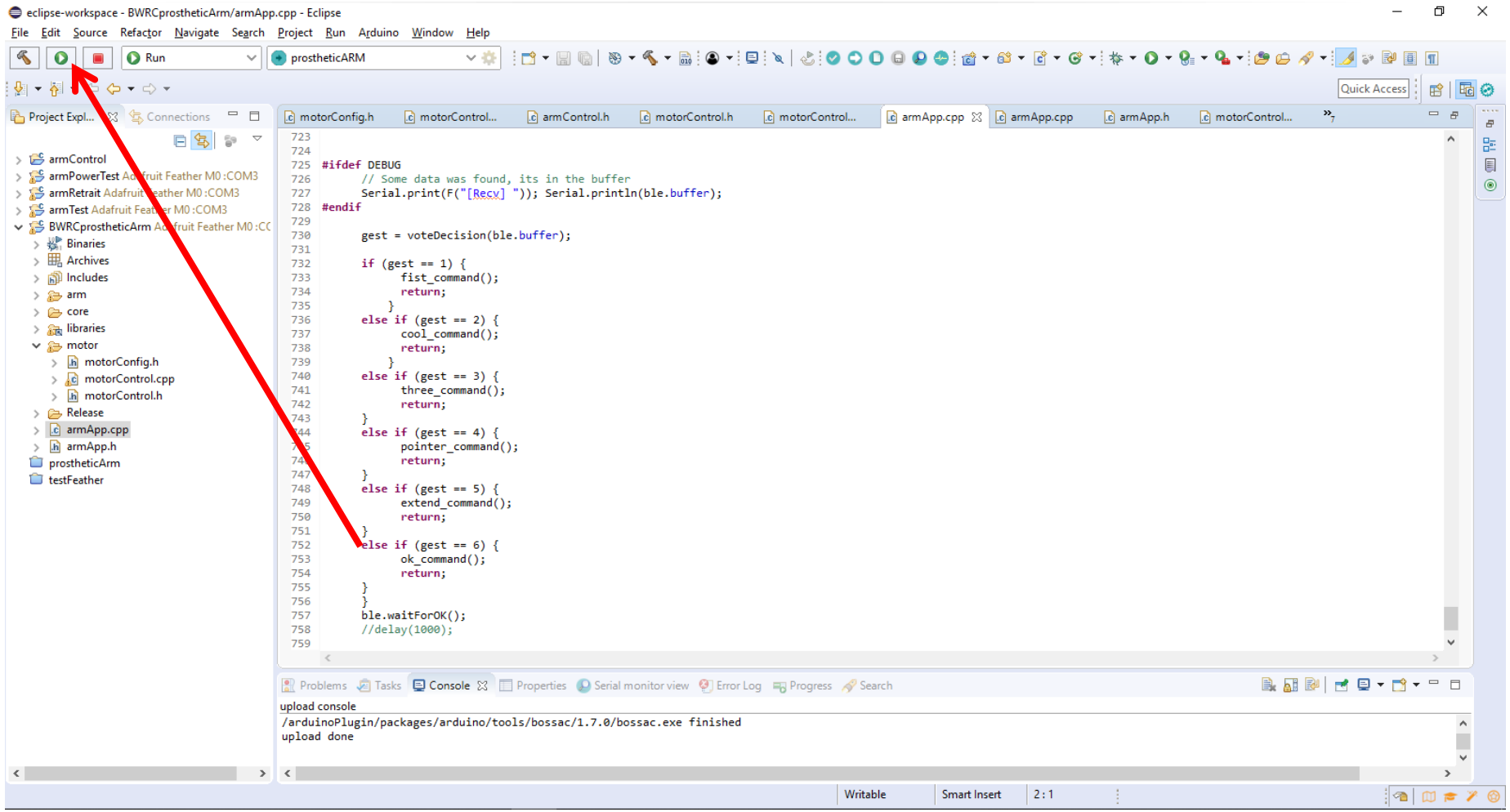
Mount – Wirst & socket assembly set up

Mount – Solder battery cable

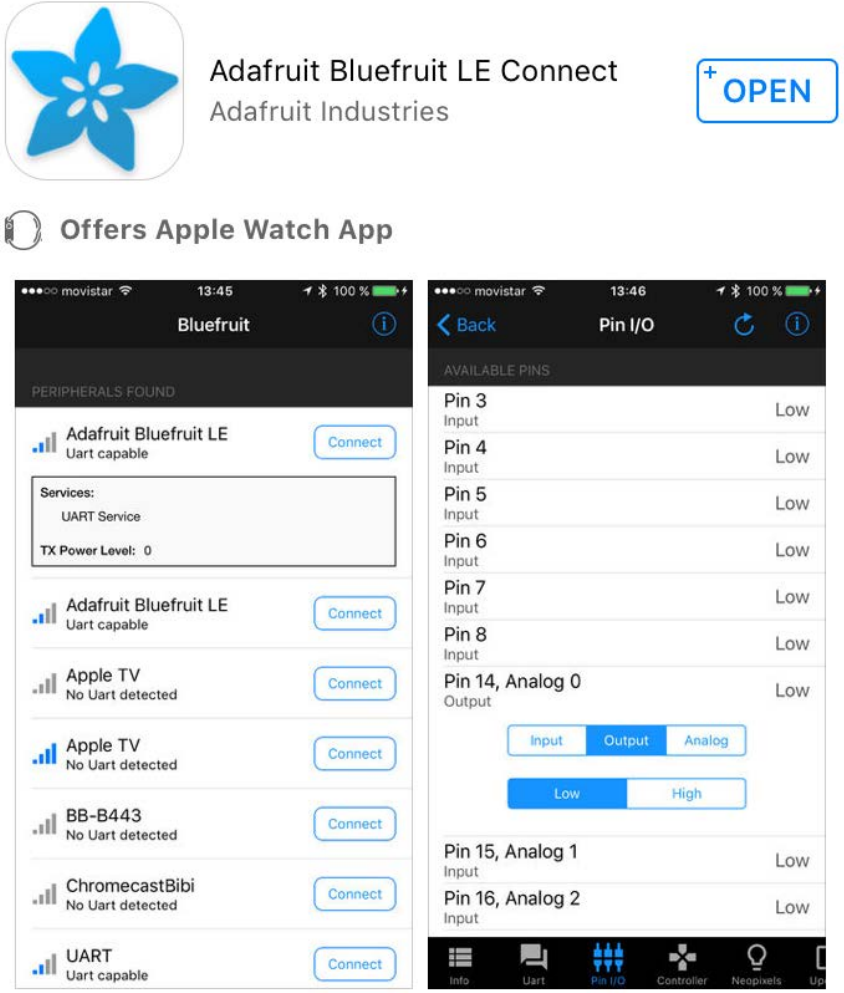
- Solder the cable as shown
- check + and - between cable and battery
- Screw the battery holder on the socket
- Put the battery on the battery holder

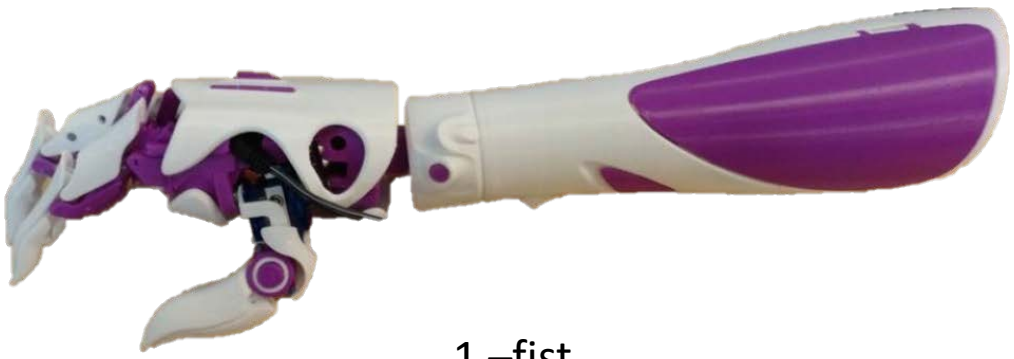


Upload code


eclipse-workspace - BWRCprostheticArm/armApp.cpp - Eclipse
File Edit Source Refactor Navigate Search Project Run Arduino Window Help
Run
prostheticARM
Quick Access
Project Expl...
Connections
armControl
armPowerTest Adafruit Feather M0 :COM3
armRetrait Adafruit Feather M0 :COM3
armTest Adafruit Feather M0 :COM3
BWRCprostheticArm Adafruit Feather M0 :CC
Binaries
Archives
Includes
arm
core
libraries
motor
motorConfig.h
motorControl.cpp
motorControl.h
Release
armApp.cpp
armApp.h
prostheticArm
testFeather
motorConfig.h
motorControl...
armControl.h
motorControl.h
motorControl...
armApp.cpp
armApp.cpp
armApp.h
motorControl...
#ifdef DEBUG
// Some data was found, its in the buffer
Serial.print(F("[Recv] ")); Serial.println(ble.buffer);
#endif
gest = voteDecision(ble.buffer);
if (gest == 1) {
fist_command();
return;
}
else if (gest == 2) {
cool_command();
return;
}
else if (gest == 3) {
three_command();
return;
}
else if (gest == 4) {
pointer_command();
return;
}
else if (gest == 5) {
extend_command();
return;
}
else if (gest == 6) {
ok_command();
return;
}
}
ble.waitForOK();
//delay(1000);
Problems Tasks Console Properties Serial monitor view Error Log Progress Search
upload console
/arduinoPlugin/packages/arduino/tools/bossac/1.7.0/bossac.exe finished
upload done
Writable
Smart Insert
2 : 1

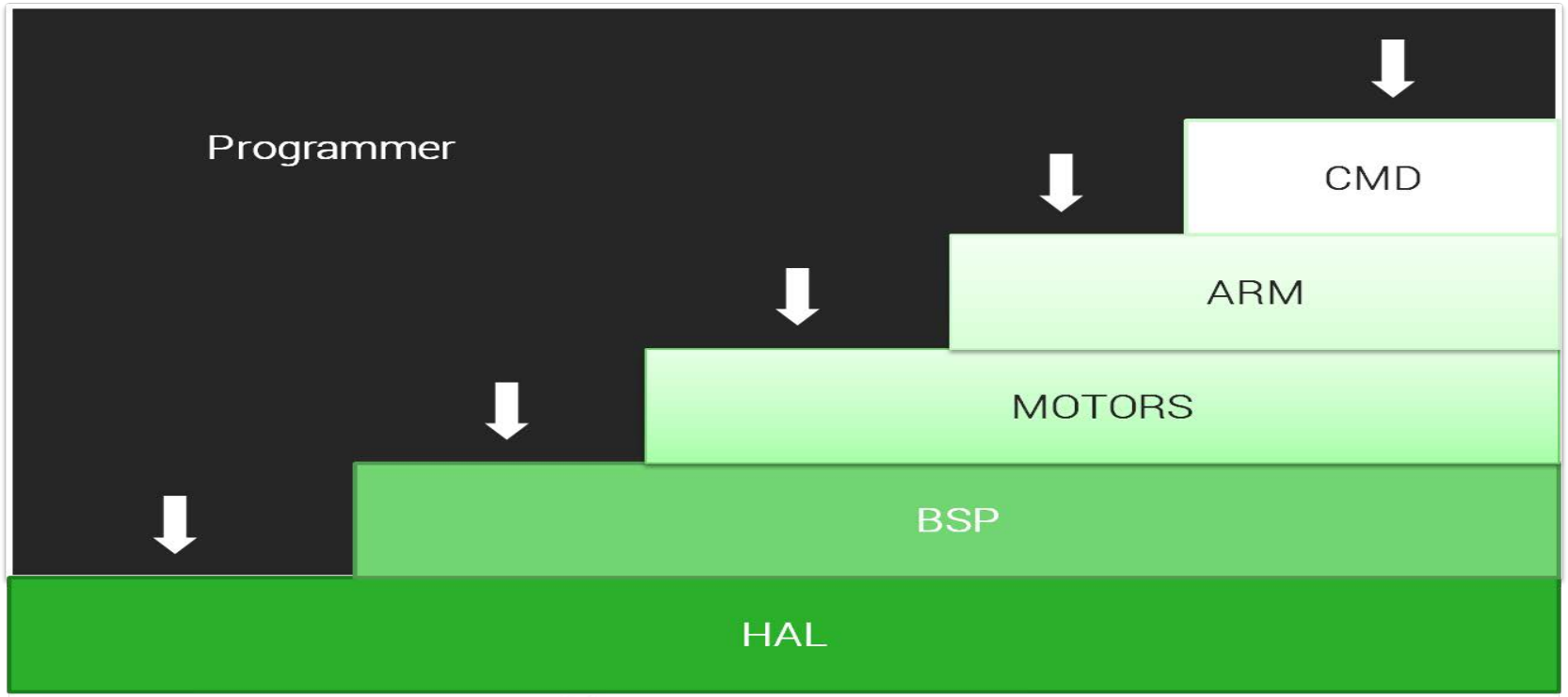
How is working now…

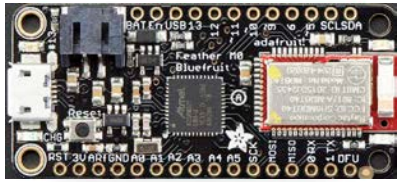




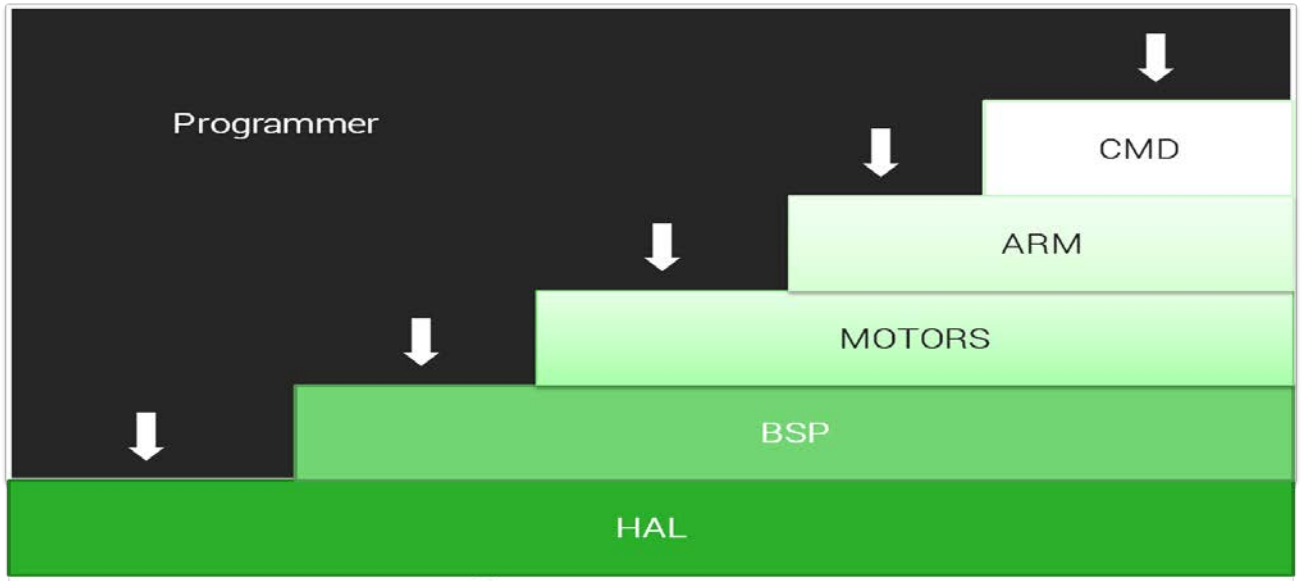
1 –fist
2-cool
3-three
4-pointer
5-extend
6-ok

Architecture





Architecture

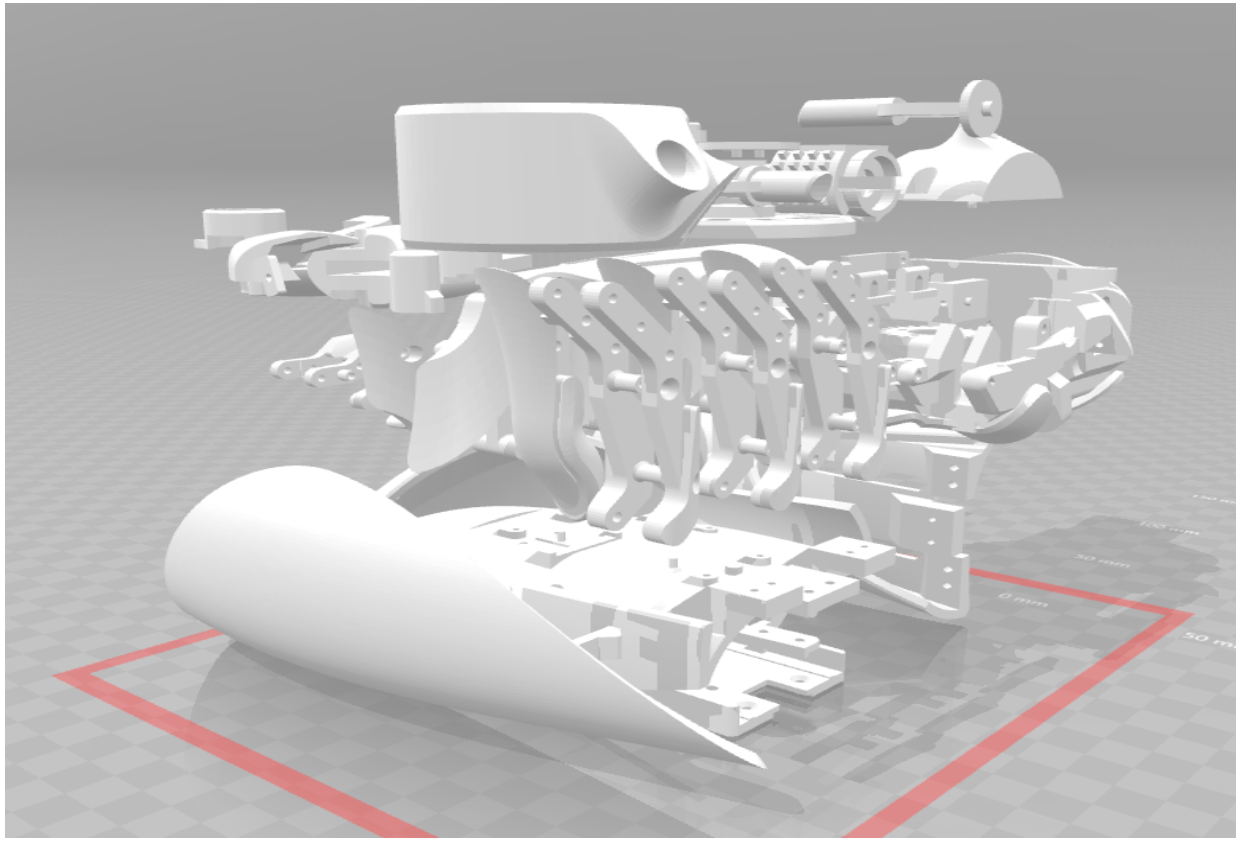


- o Independent
- o Arm hardware dependent
- o Motor dependent
- o Platform/OS dependent
- o µP dependent

- o Electronics + Mechanical

Mechanical

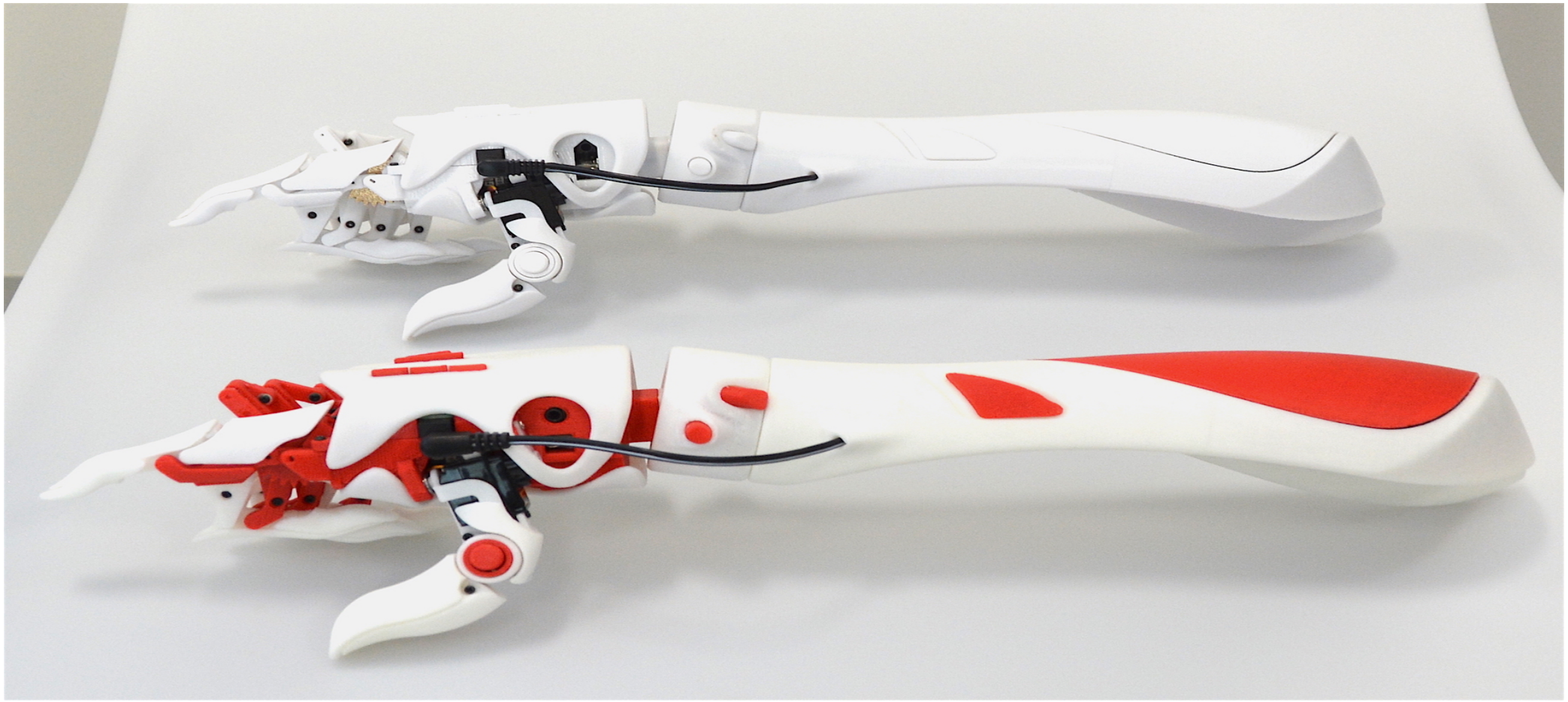
- Exiii based (GPL and Creative Commons licenses)



- http://exiii-hackberry.com/

Mechanical

- Exiii based



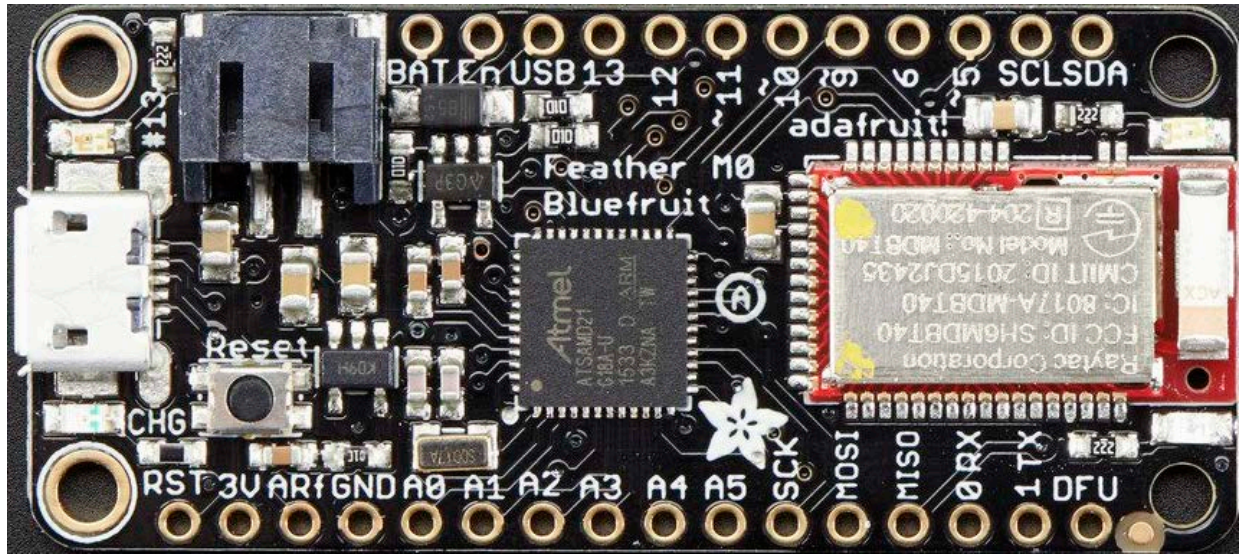
Mechanical

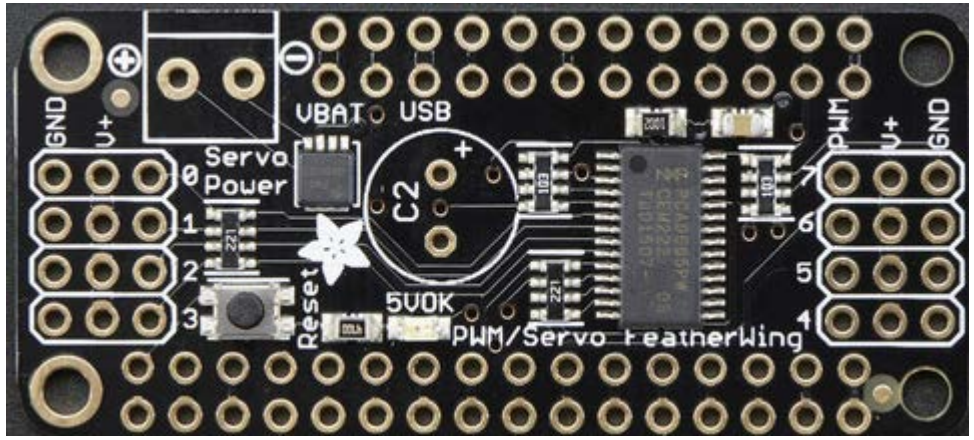
- Motors
  - Index
  - Thumb
  - Fingers
  - Wrist – elevation
  - Wrist – azimut

- Sensor
  - Accelerometer XYZ (feedback)

Electronics

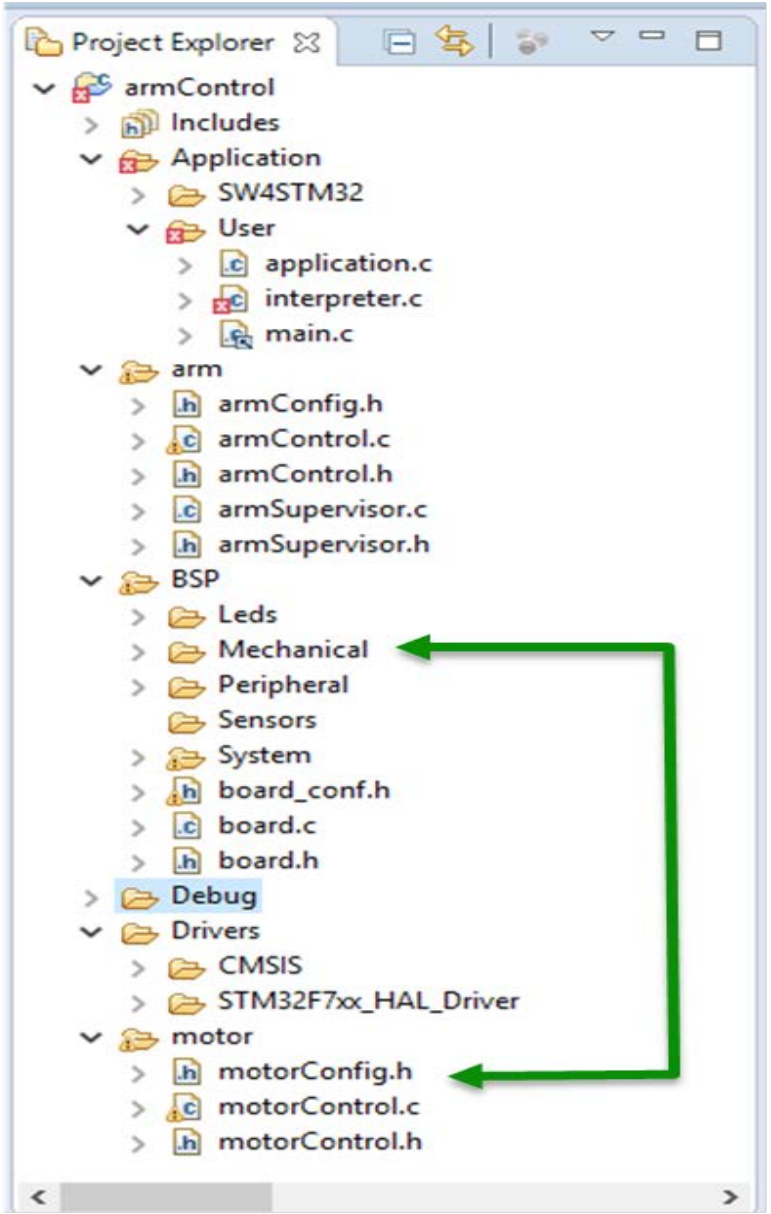
- Cortex M0
- BLE
- Feather Platform
- Arduino tools compatible
- PWM 8 channels





SW architecture

- Application
- Arm
- motor
- BSP
- Drivers/HAL



Motor

- motorConfig.h
- motorControl.h
- motorControl.c

Motor

- motorConfig.h

```
#define MOTORCONTROL_HOMING_TIMEOUT 300 /* seconds */

const motorControlConfig_t motorControlConfig_thumb =
{
    .limitMode = MOTORLIMITS_BOTHSWITCHS,
    .limitWarningZone = 0, //Microsteps
    .limitEnd = 0
};
```

Motor

- motorControl.h

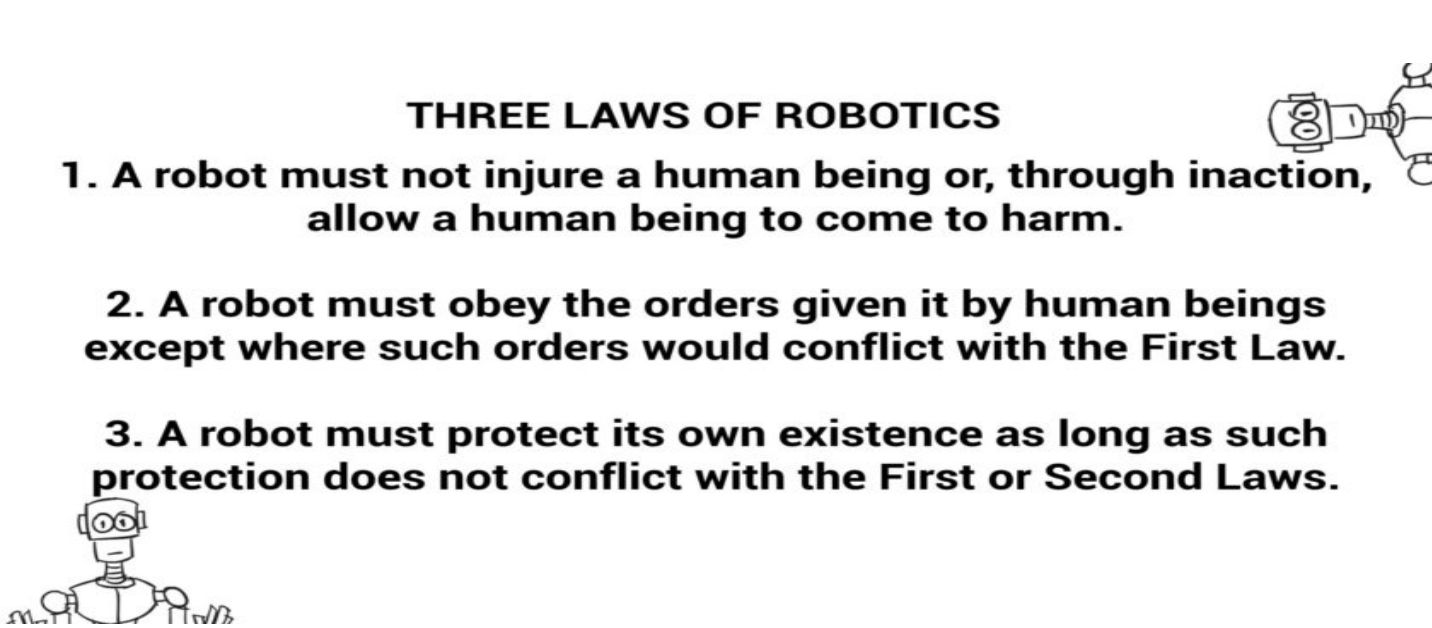
```
typedef struct motorControlConfig_
{
    motorLimits_t limitMode;
    int32_t limitWarningZone;
    int32_t limitEnd;
}motorControlConfig_t;

typedef enum{
    MOTORCONTROL_ENABLED,
    MOTORCONTROL_DISABLED,
}motorControl_status_t;

typedef struct motorUnit_
{
    tmcm1lx0_t* tmcm_driver;
    motorStatus_t status;
    motorOperation_t* operation;
    tmcm1lx0_config_t* config;

    motorControlConfig_t* controlConfig;
    uint8_t shouldBeHomed;
    uint32_t measuredDistanceBetweenSwitches;

}motorUnit_t;

typedef struct motorControl_
{
    motorControl_status_t status;
    gen_list* motors;
```

Motor

- motorControl.h
  - **motorControl_init** (motorControl_t** motorControl);
  - **motorControl_addMotor** (motorControl_t* motorControl, motorUnit_t** motorUnit, tmcm11x0_t* tmcm_motor, tmcm11x0_config_t* motorConfig, motorControlConfig_t* controlConfig);
  - **motorControl_start** (motorControl_t* motorControl);
  - **motorControl_stop** (motorControl_t* motorControl);
  - **motorControl_unit_moveSteps** (motorControl_t* motorControl, motorUnit_t* motorUnit, motorDirection_t direction, uint32_t steps);
  - **motorControl_unit_position**();
  - **motorControl_unit_setPositioningSpeed**();

Arm

- armConfig.h
- armControl.h
- armControl.c
- armSupervisor.h
- armSupervisor.c



Arm

- armConfig.h

```
/* Relation from motor to wrist (motor->gears->linear) */
#define WRIST_CIRCULAR_REDUCTION 256.0          /* Adimensional */
#define WRIST_CIRCULAR_FINALRELATION 1.0            /* Adimensional */
#define WRIST_CIRCULATOLINEAR_CONVERSION 15.0   /* cm/revolution */

/* All different relations between motors and arm */


/* Fingers relations (motor->gears->wormdrive->fingers) */
#define FINGERS_CIRCULAR_REDUCTION 64.0         /* Adimensional */
#define FINGERS_CIRCULARTOLINERAR_CONVERSION 5  /* mm/revolution */
#define FINGERS_STROKE_LENGTH 25                    /* mm */
#define FINGERS_AUTOADJUST_SPEED 25 /*Standar speed*/

/* Waterjet relations (motor->gears->angle) */
#define WRIST_AZIMUTH_CIRCULAR_REDUCTION 30.0       /* Adimensional */
#define WRIST_AZIMUTH_CIRCULAR_FINALRELATION 1.0    /* Adimensional */
#define WRIST_ELEVATION_CIRCULAR_REDUCTION 30.0     /* Adimensional */
#define WRIST_ELEVATION_CIRCULAR_FINALRELATION 1.0  /* Adimensional */
```

Arm

- armControl.h

```
//Wrist definitions

typedef enum {
    WRIST_UP,
    WRIST_DOWN,
    WRIST_LEFT,
    WRIST_RIGHT,
}armWristDirection_t;

typedef enum {
    WRIST_VIEWLIMIT_LOWER,
    WRIST_VIEWLIMIT_UPPER,
}armWristViewLimit_t;

typedef struct armWrist_{
    armMotorChannel_t* azimuth;
    armMotorChannel_t* elevation;
    int32_t azimuthLowerLimit;
    int32_t azimuthUpperLimit;
    int32_t elevationLowerLimit;
    int32_t elevationUpperLimit;
}armWrist_t;

typedef enum {
    WRIST_OP_STOPPED,
    WRIST_OP_MOVING,
    WRIST_OP_ADJUSTING,
    WRIST_OP_TIMEOUT,
}armWristOperationStatus_t;

typedef enum {
    WRIST_DIR_UP,
    WRIST_DIR_DOWN,
    WRIST_DIR_LEFT,
```

Arm

- armControl.h
  - **armControl_create** (armControl_t** armControl);
  - **armControl_configure** (armControl_t* armControl, motorControl_t* motorControl, motorUnit_t* fingers, motorUnit_t* index, motorUnit_t* thumb, motorUnit_t* wristAzimuth, motorUnit_t* wristElevation);
  - **armControl_getAcceleration**();
  - **armControl_wrist_move** (armControl_t* armControl, armWristDirection_t direction, float32_t speed);
  - **armControl_wrist_position**();
  - **armControl_wrist_stop();**
  - **…**

Application

- Application.c
  - Init
  - Configuration
  - Wait for orders (TxRx/UART…)
  - Orders -> Arm Commands

Alvaro Araujo
araujo@b105.upm.es
araujo@berkeley.edu

BWRC rocks!!