

Apple HomeKit Garage Door Opener as an IoT Device

release no releases or repo not found



See also [Project README](#) for more information.

This example shows a garage door opener that can be controlled by Apple HomeKit. The device uses the same function as a traditional button to open the door. An ultrasonic distance sensor monitors the position of the door to tell Apple HomeKit whether the door is closed, open, or half open.

New or different functions can be configured via CHost, derived from CUnitBase, so that only a small amount of programming is required, e.g. to detect the door position via limit switches or to start the door motor other than by a pulse.

Installation of the box and wiring of the door operator

The project can be used to control garage doors of this type.



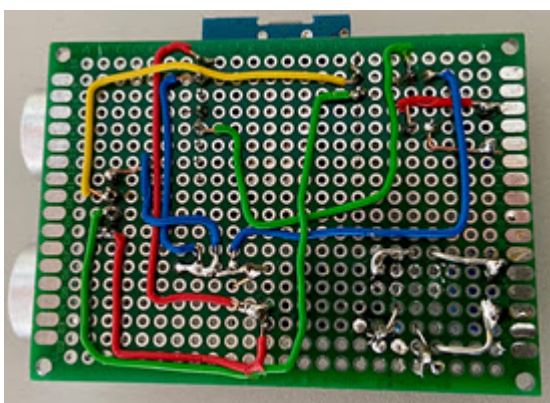
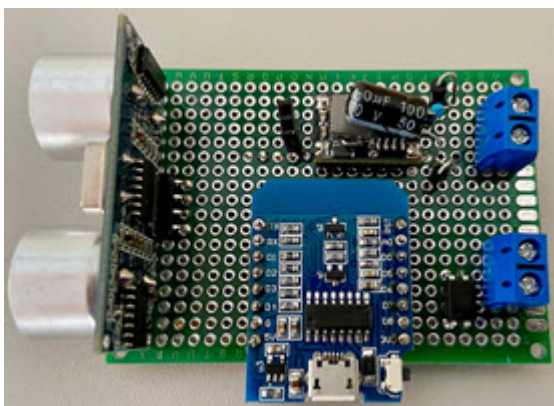
The garage door opener is installed in a black box (left) and connected to the door operator.



Power is supplied from the AC 24V provided by the door operator. As standard the door is operated by a push button. A short pulse is sent to the door operator. The same cable is also connected to the IoT device.

! Important: The ultrasonic distance sensor must be aligned so that the top of the door is detected when the door is closed and the maximum distance is measured when the door is closed.

IoT device with PCB



Compile and upload

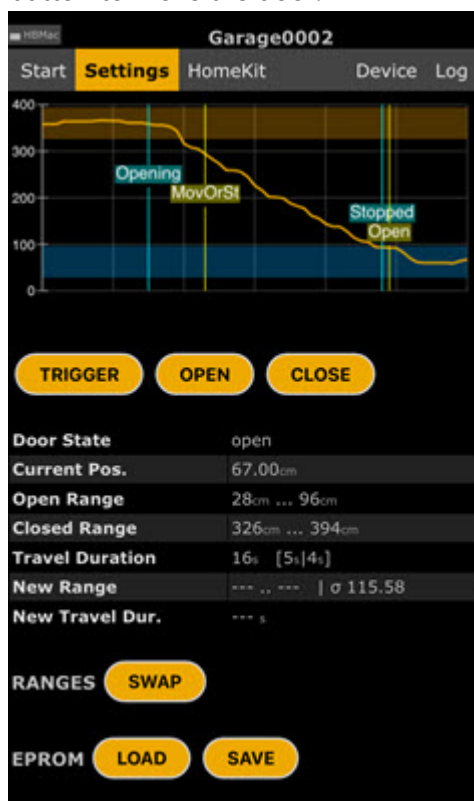
The configuration of the garage door opener is done in the file `GarageDoor.h`. The following settings are available:

```
/*
 * @brief This pin is used to start the door motor by pulling the trigger pin HIGH
 * for 100 ms.
 * @note Make sure that this pin is set to LOW during booting so that
 *       no trigger pulse is sent when restarting.
 *       For example: D2 would be possible, but not D3.
 */
#define DOOR_PIN  D2

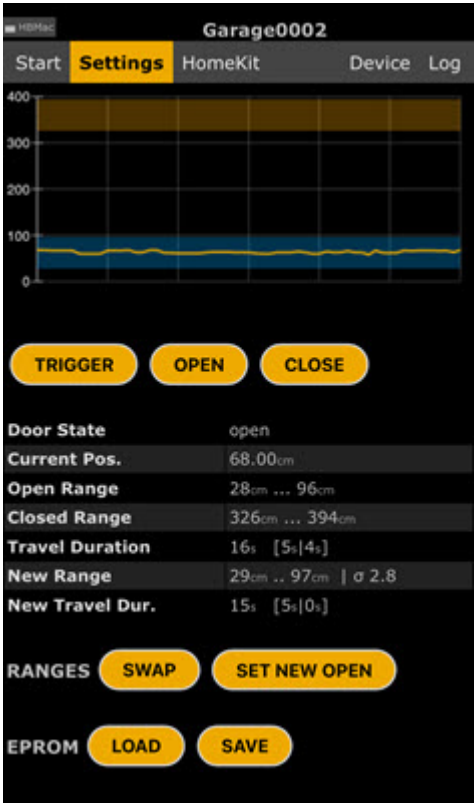
/* @brief The unit reads the door position from an ultrasonic sensor.
 * The sensor is mounted at the top of the door and measures the distance to the
 * door.
 * @note Sensor: HC-SR04 Ultrasonic Sensor
 * @note The EchoPin must be connected to an interrupt pin. (e.g.: D8)
 */
#define TRIGGER_PIN D7
#define ECHO_PIN    D8 // must be interrupt capable
```

Start-up and configuration of the device

1. The first step is to locate the closed and open positions of the door. To do this, press the **OPEN** or **CLOSE** button to move the door.



2. After reaching the end position, wait until a flat line appears on the graph. The **SET NEW OPEN** or **SET NEW CLOSE** button will appear. Press this button to save the current position as **OPEN** or **CLOSE**. Repeat the procedure for the other direction to set the **OPEN** and **CLOSE** positions.



3. Now press the **SAVE** button.

- ! Captured position data must be stored permanently.
- 💡 Reverse **OPEN** and **CLOSE** can be corrected using the **SWAP** button.
- 💡 The **SAVE** button permanently saves the current information and the **LOAD** button loads the last saved items.