Engineering Technology (ENGR 101)

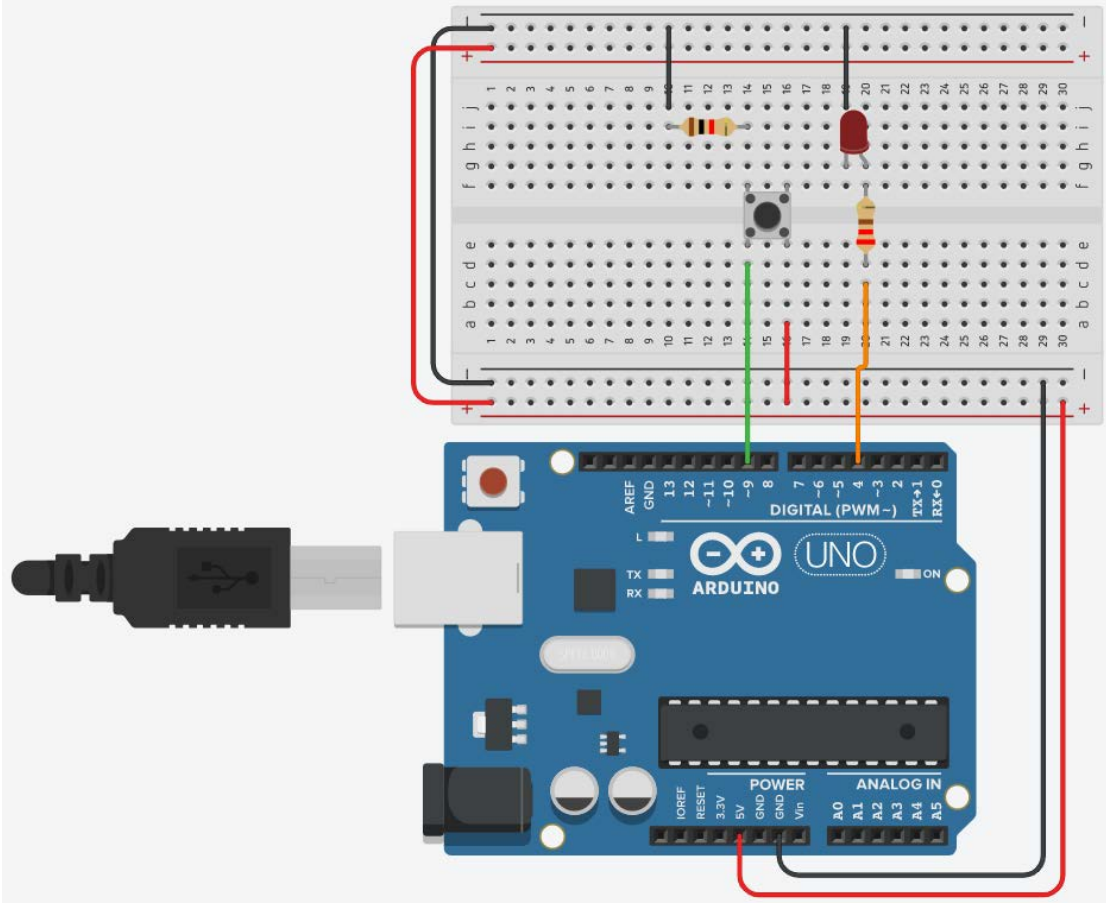
Finite State Machine (FSM)

Pushbutton Switch

- Turning a LED on and off using a pushbutton switch

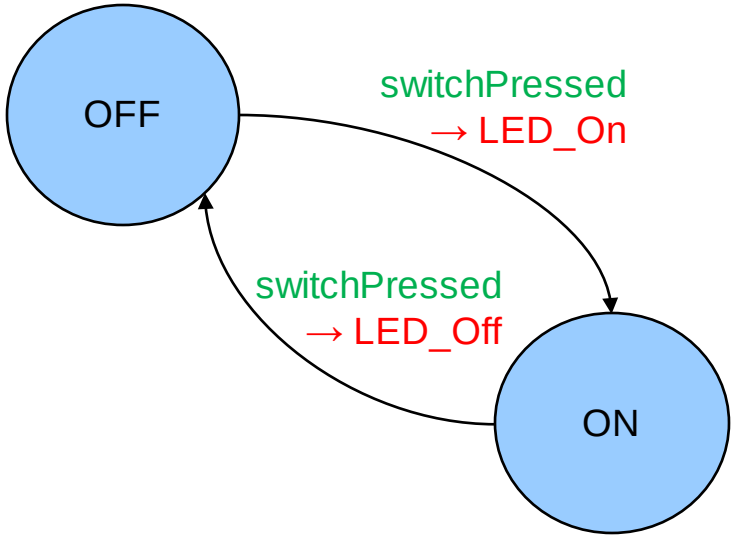
```
int switchPressed = 0;

void loop() {
    // Get the input
    switchPressed = digitalRead(switchPin);
    if(switchPressed == HIGH){
        delay(50); // Debounce time of 50ms
        switchPressed = digitalRead(switchPin);
        if(switchPressed == HIGH){
            // Change the output
            digitalWrite(LEDpin, HIGH);
        }
    }
    else if (switchPressed == LOW ) {
        digitalWrite(LEDpin, LOW);
    }
}
```



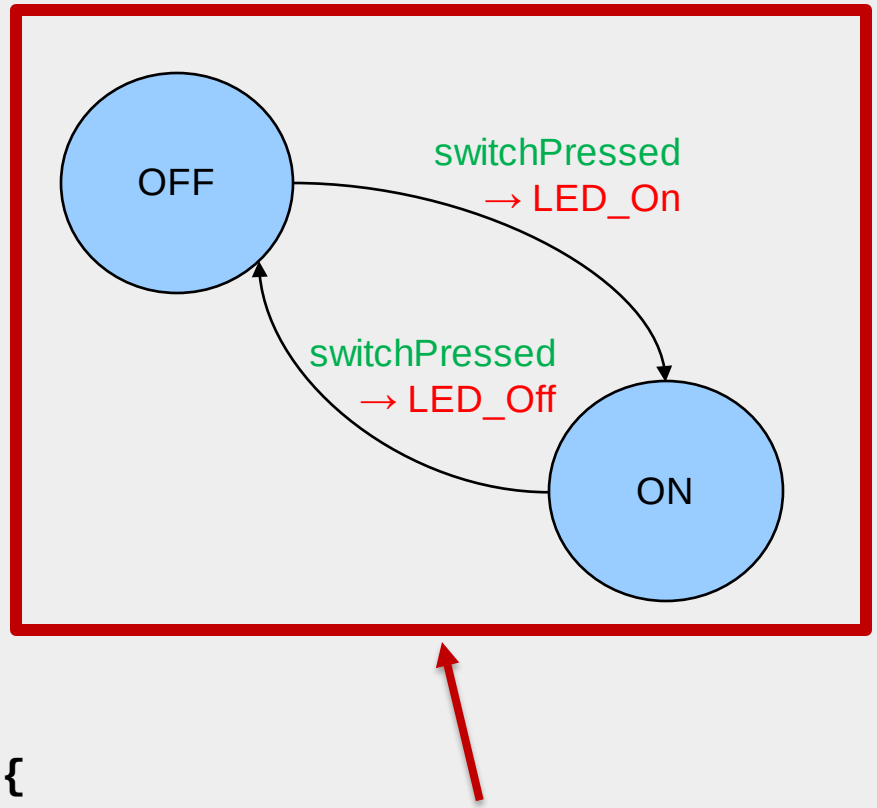
Pushbutton Switch tweaking

- In the previous sketch, LED is on while the switch is pressed and LED is off while the switch is not pressed.
- How can we turn the LED on or off only when the switch is pressed?
  - We will need to remember the states of the switch and LED.



Pushbutton Switch tweaking

```
int ledPin = 4;
int switchPressed = 0;
String state =  “OFF”;
void loop() {
    //get the input
    switchPressed = digitalRead(switchPin);
    if(switchPressed == HIGH){
       delay(50); // debounce time of 50ms
       switchPressed = digitalRead(switchPin);
       if(switchPressed == HIGH){
          //Transitions to new state and output
          if (state == “OFF” && switchPressed == HIGH){
             state = “ON”;
             digitalWrite(ledPin, HIGH);
          }
          else if (state = “ON” && switchPressed == HIGH){
             state = “OFF”;
             digitalWrite(ledPin, LOW);
          }
       }
    }
}
```



Finite State Machine

What are FSM's?

- A way of thinking about certain kinds of problems
- A way of describing/modelling/analysing systems
- A way of designing solutions to a wide range of engineering problems.

- Example applications
  - Controllers for physical devices
  - Protocols for communications/networking
  - Regular expression searching in text
  - Analysing and Designing user interfaces.
  - Speech recognition
  - ….

States and Transitions

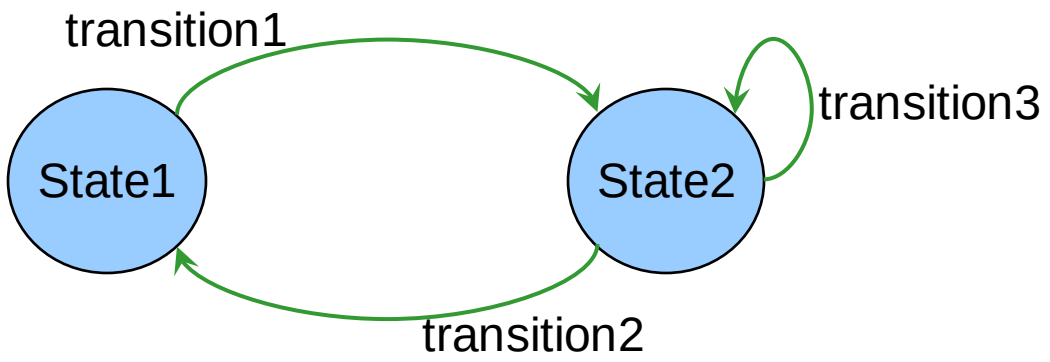
Finite means limited in size. Opposite of Infinite ∞.

State is the condition or remembered information of a system.

- Key idea:
  - If a system has finite set of possible states,
  - then you can list them and
  - identify each of the possible things (Arduino inputs) that could happen in each state.
    - These possible Arduino inputs can change one state to another state called a *transition*

Transition when something changes from one state to another.

- Typically:
  - draw labelled circles for the states
  - draw labelled arrows for the transitions between them



Kinds of Finite State Machines.

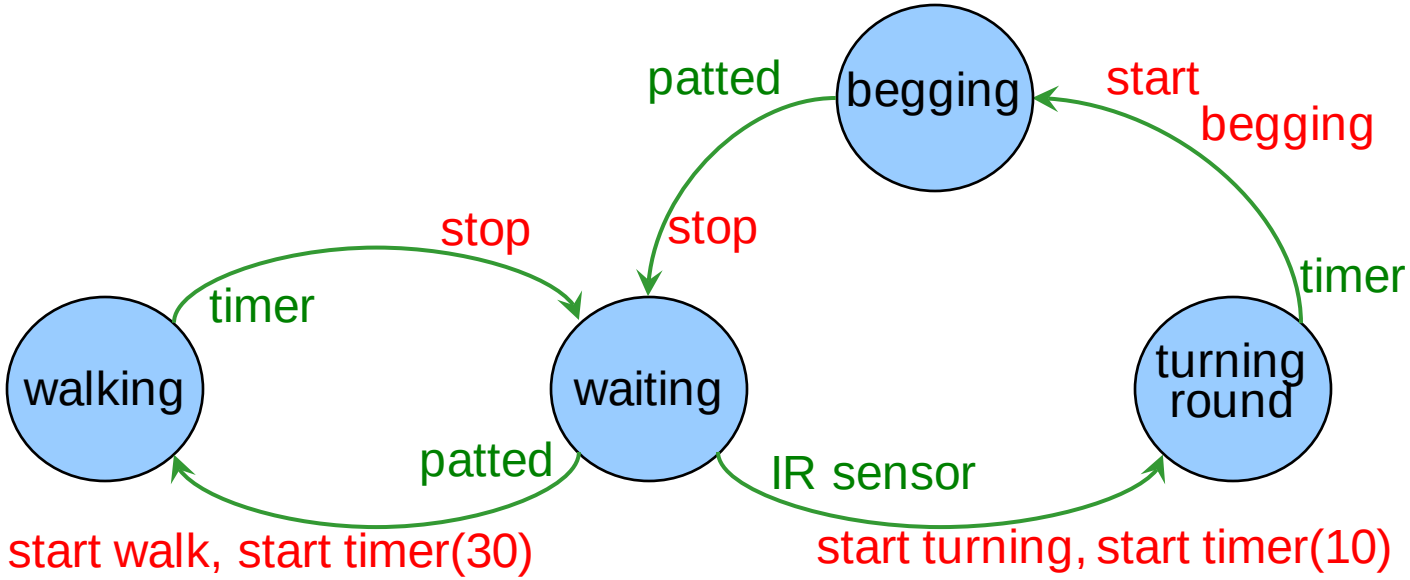
- Describing/specifying an active system
  - states the Arduino sketch can be in
  - input/signals/sensors that the Arduino may respond to
  - actions that the Arduino will perform

Example: an autonomous vehicle with sensors

a controller for a robotic toy (For example: *Auti* is a toy for autistic children)

a controller for a traffic light (Lab 3)

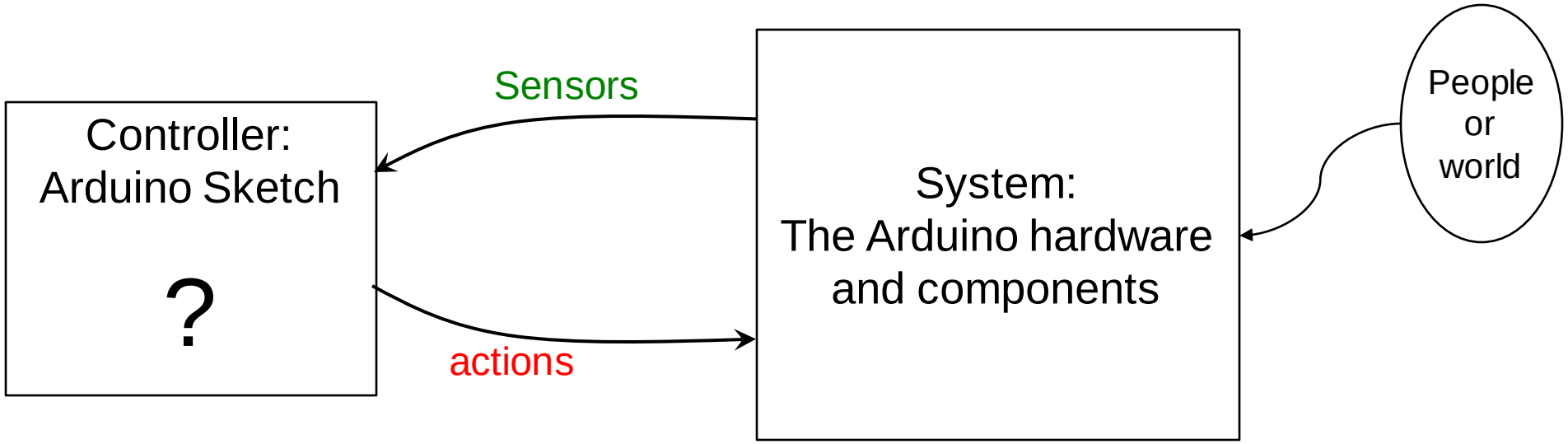
…



Auti designed by
Helen Andreae

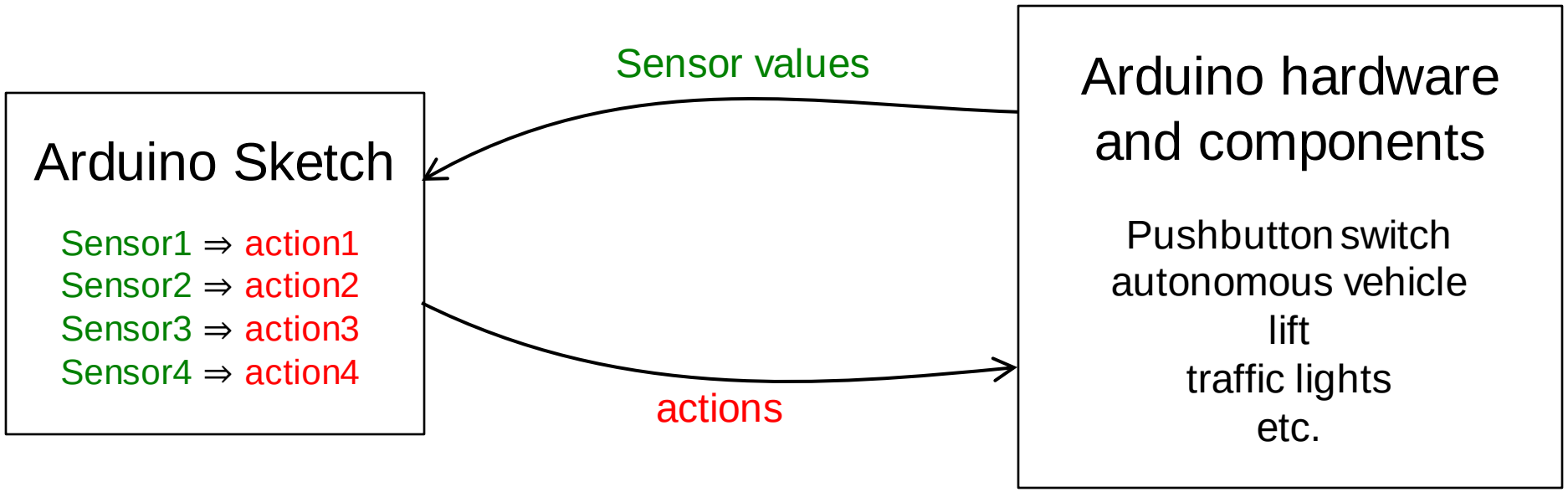
Controllers - Arduino

- Devices to control systems.
  - Could be hardware, but usually software, might be a microprocessor or a large computer.
  - Controller (Arduino Sketch) gets input from the system
  - Controller (Arduino Sketch) performs actions on the system to change its behaviour

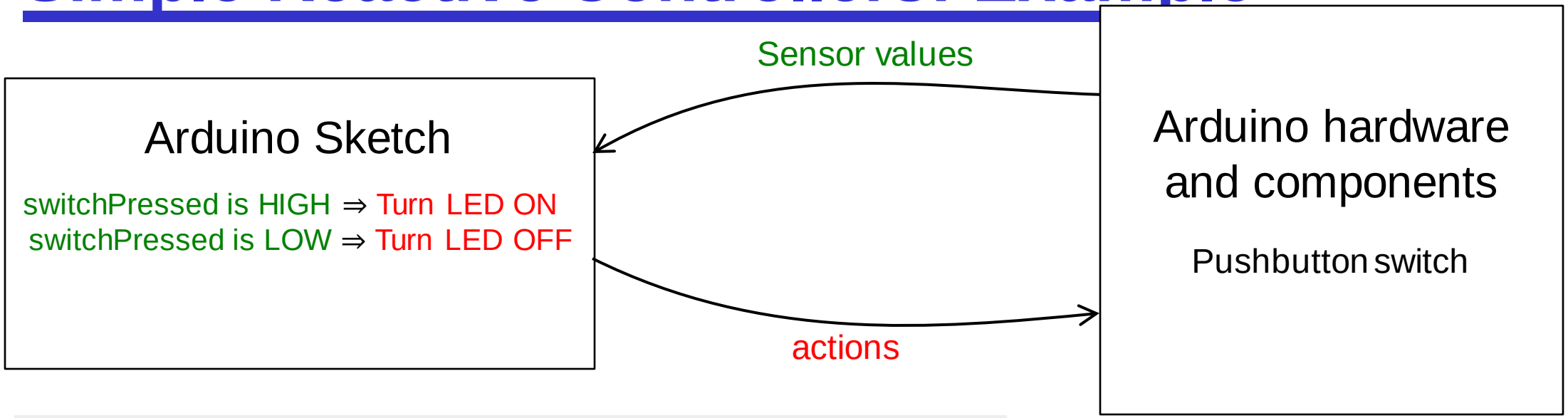


Simple Reactive Controllers.

- Arduino Sketch drives the hardware and components:
  - Input: Signal/sensor value from the hardware and components triggers an action
  - Output: Action, reacts to the signal
  - Same response to a sensor, every time
  - Very limited



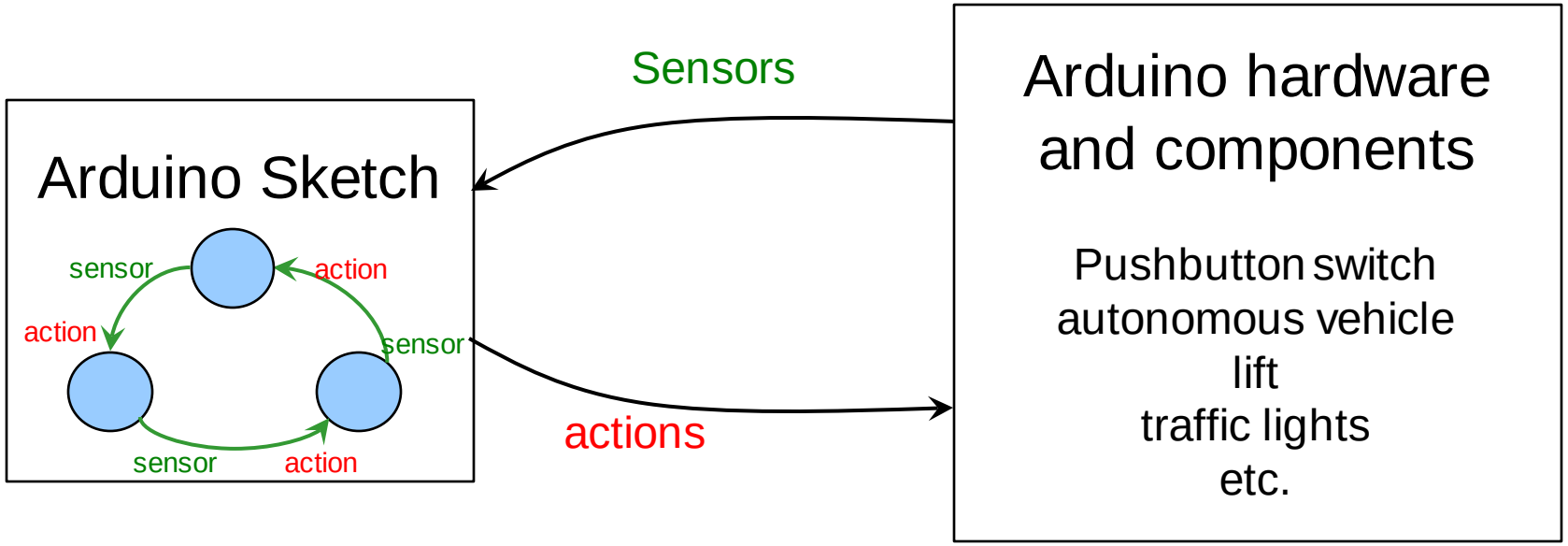
Simple Reactive Controllers: Example



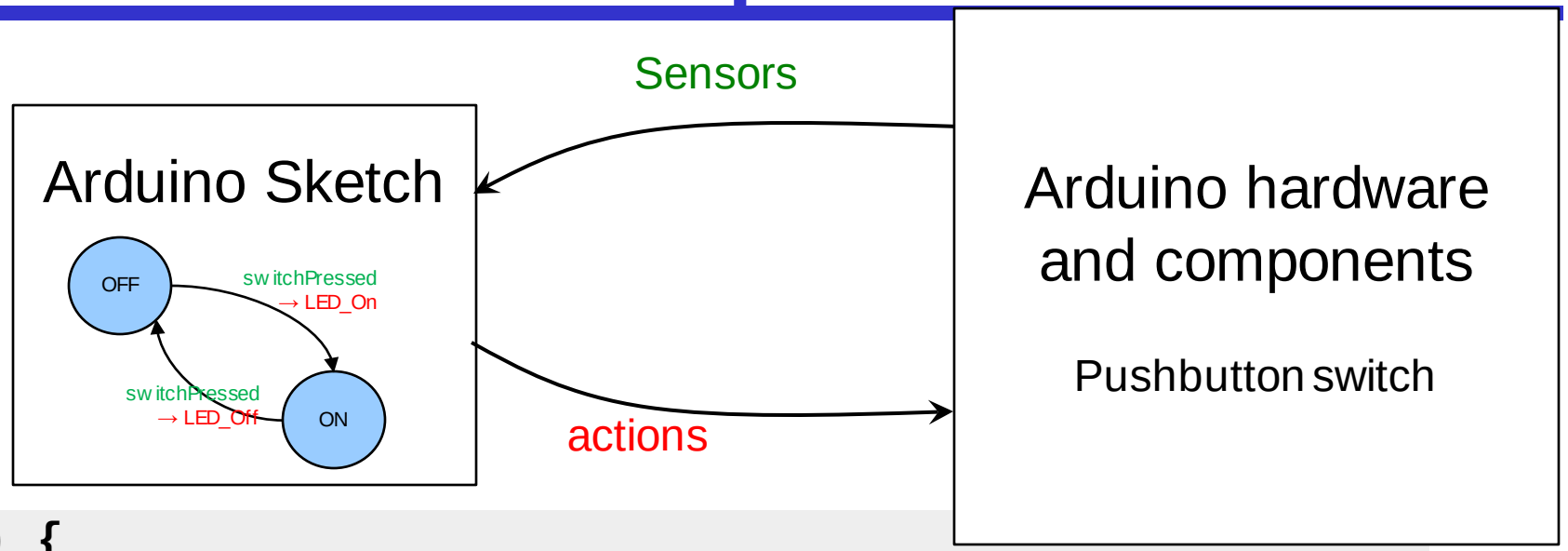

```
void loop() {
    // Get the input
    switchPressed = digitalRead(switchPin);
    // Change the output
    if (switchPressed == HIGH ) {
       digitalWrite(LEDpin, HIGH);
    }
    else if (switchPressed == LOW ) {
       digitalWrite(LEDpin, LOW);
    }
}
```

FSM Controllers.

- Arduino Sketch drives the hardware and components:
  - Input Signal/sensor from the hardware and components
    - triggers an action
    - and changes state (of the Arduino Sketch)
  - States represent the history of what has gone before to enable different responses to same sensor at different times
  - Much richer behaviour.



FSM Controllers: Example

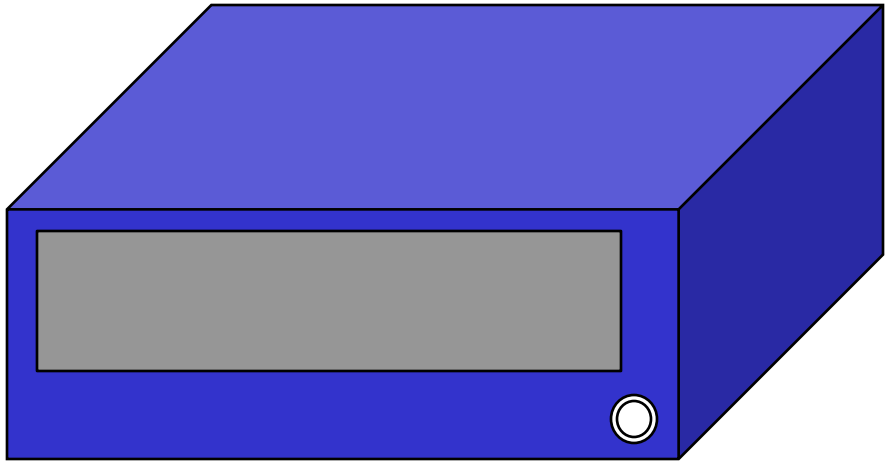



```
void loop() {
    // Get the input
    switchPressed = digitalRead(switchPin);
    // Change the output
    if (switchPressed == HIGH && state == “OFF” ){
        digitalWrite(ledPin, HIGH);
        state = “ON”;
    }
    else if (switchPressed == HIGH && state = “ON”){
        digitalWrite(ledPin, LOW);
        state = “OFF”;
    }
}
```

Example: Document Holder with Arduino

- Document holder with a fingerprint lock, and a document sensor

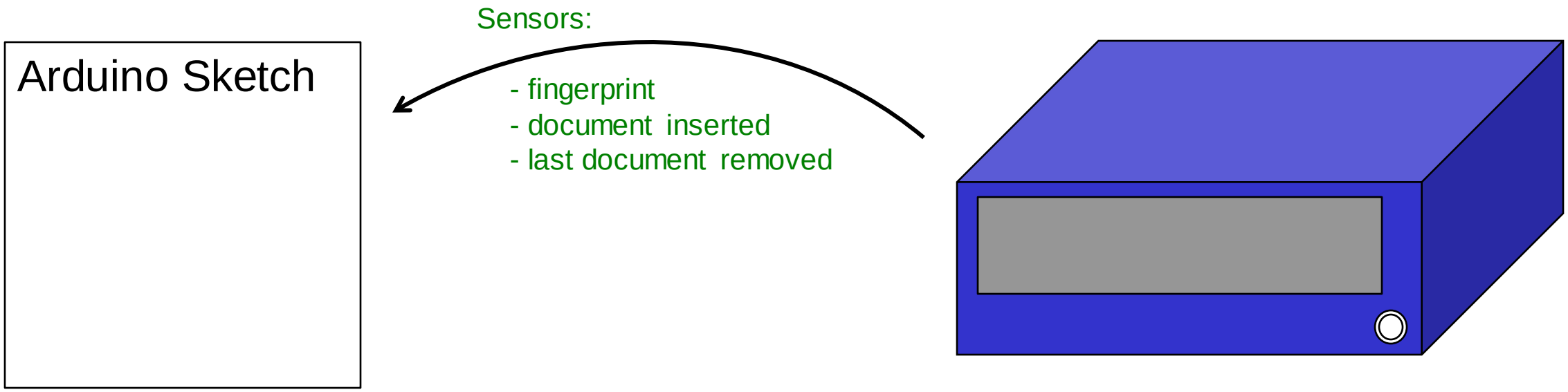
- It has to be unlocked to put documents in or out.
- If there are no documents in it, it stays unlocked.
- If there are documents in it, then the fingerprint lock will lock/unlock the holder



Example: Document Holder with Arduino

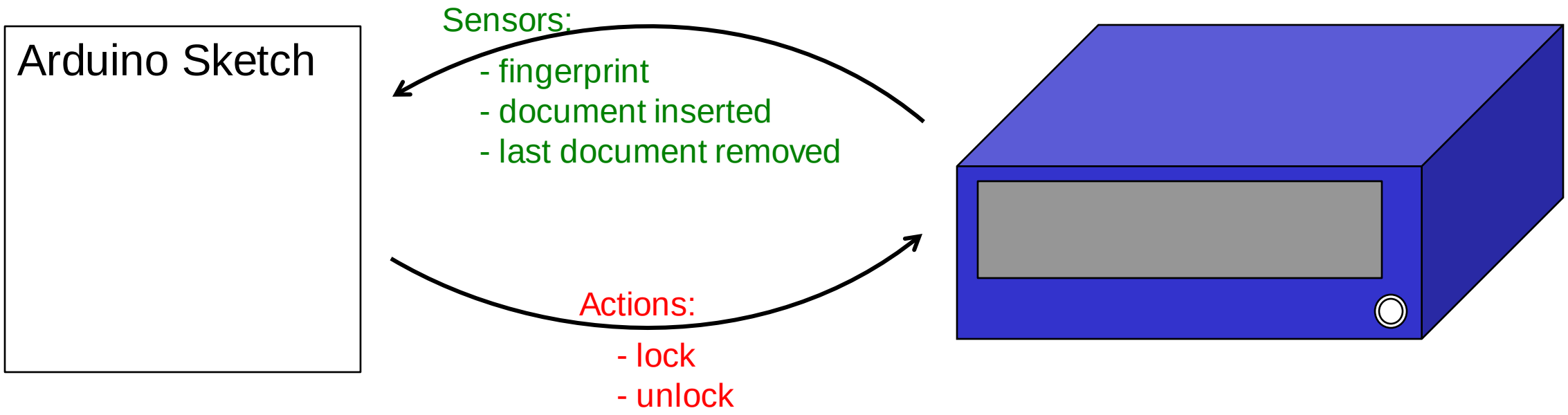
- Document holder with a fingerprint lock, and a document sensor

- It has to be unlocked to put documents in or out.
- If there are no documents in it, it stays unlocked.
- If there are documents in it, then the fingerprint lock will lock/unlock the holder



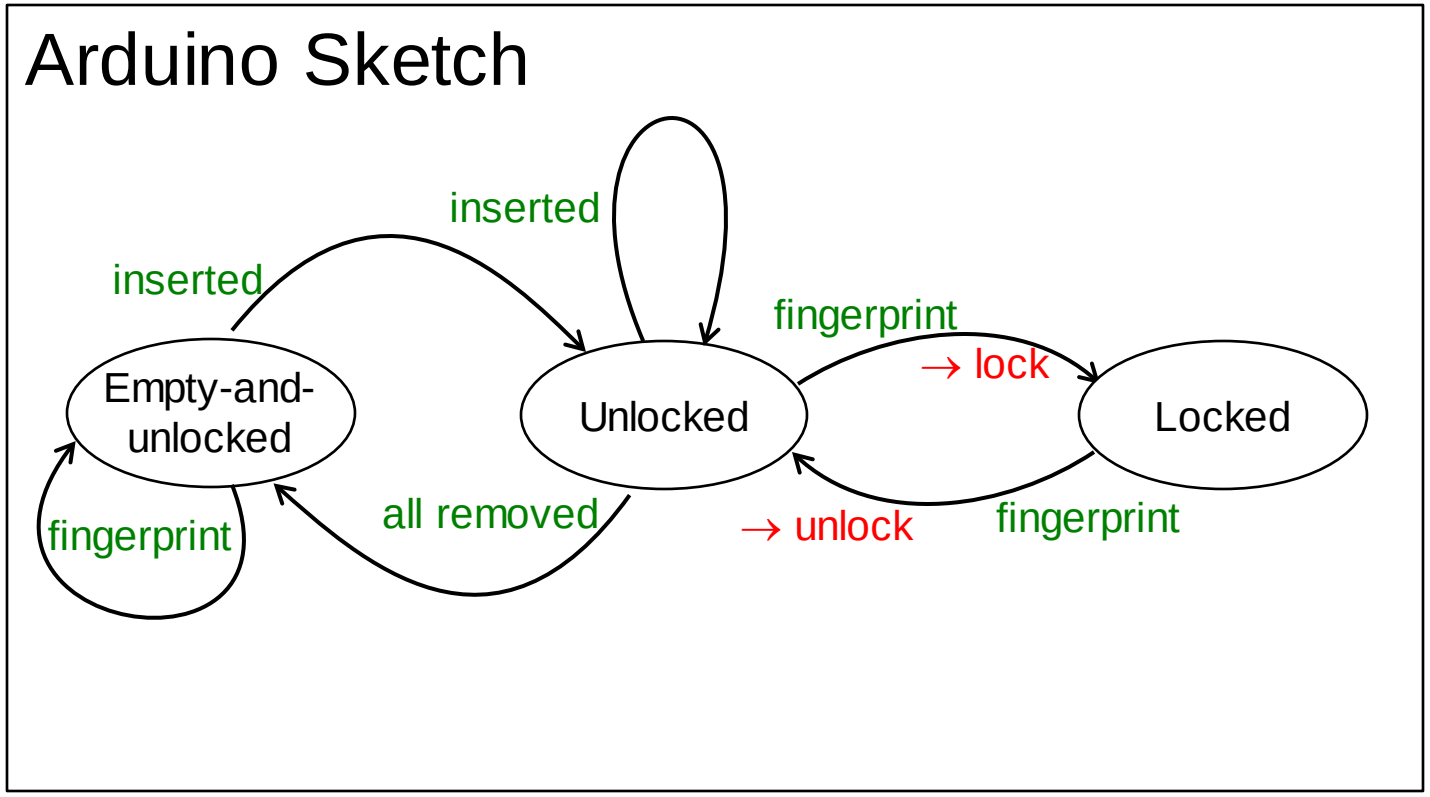
Example: Electronic Document Holder

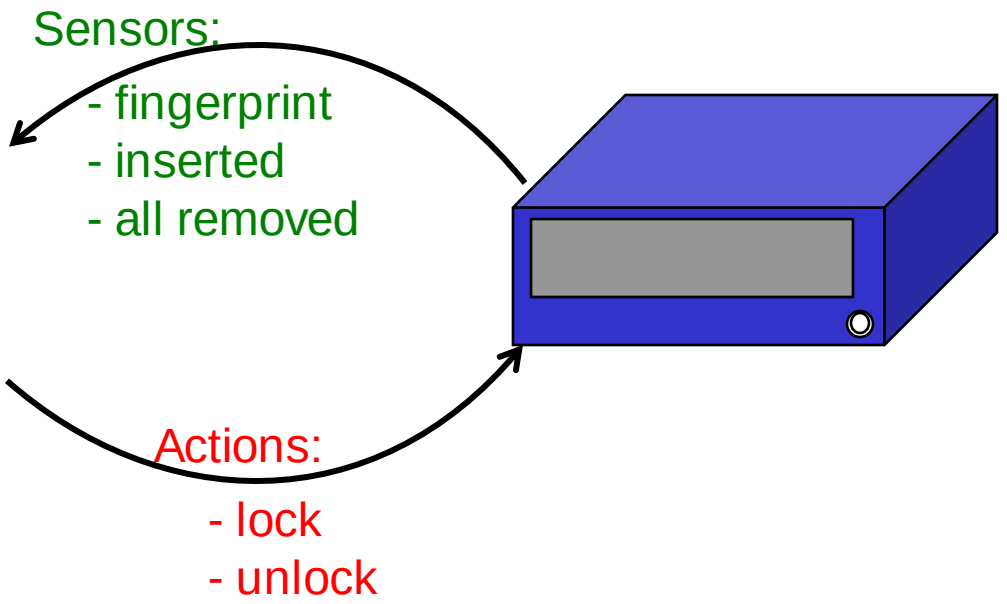
- Document holder with a fingerprint lock, and a document sensor
  - It has to be unlocked to put documents in or out.
  - If there are no documents in it, it stays unlocked.
  - If there are documents in it, then the fingerprint lock will lock/unlock the holder



Example: Electronic Document Holder

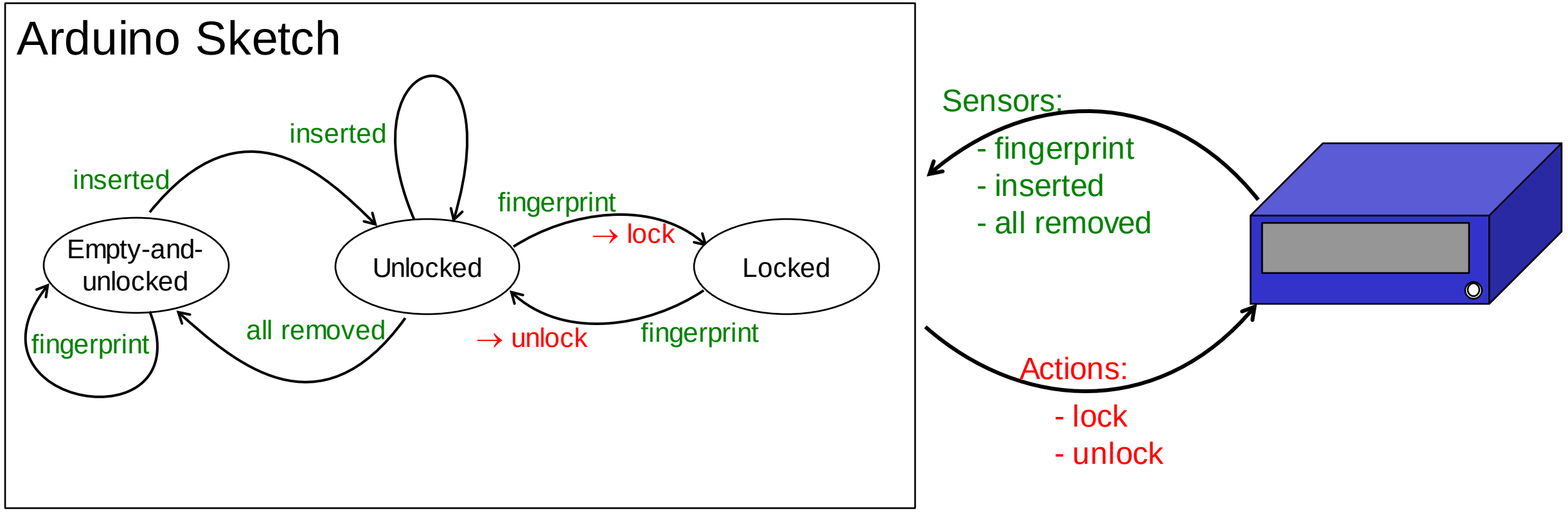
- Designing the Arduino Sketch:
  - start in one state
  - identify sensors that might happen
  - work out what the action and the new state should be.





Example: Electronic Document Holder

- Understanding the Arduino Sketch:
  - Follow the path through the states, guided by the sensor values from the Arduino hardware and components.
  - see what actions get done on the system as you follow the path.

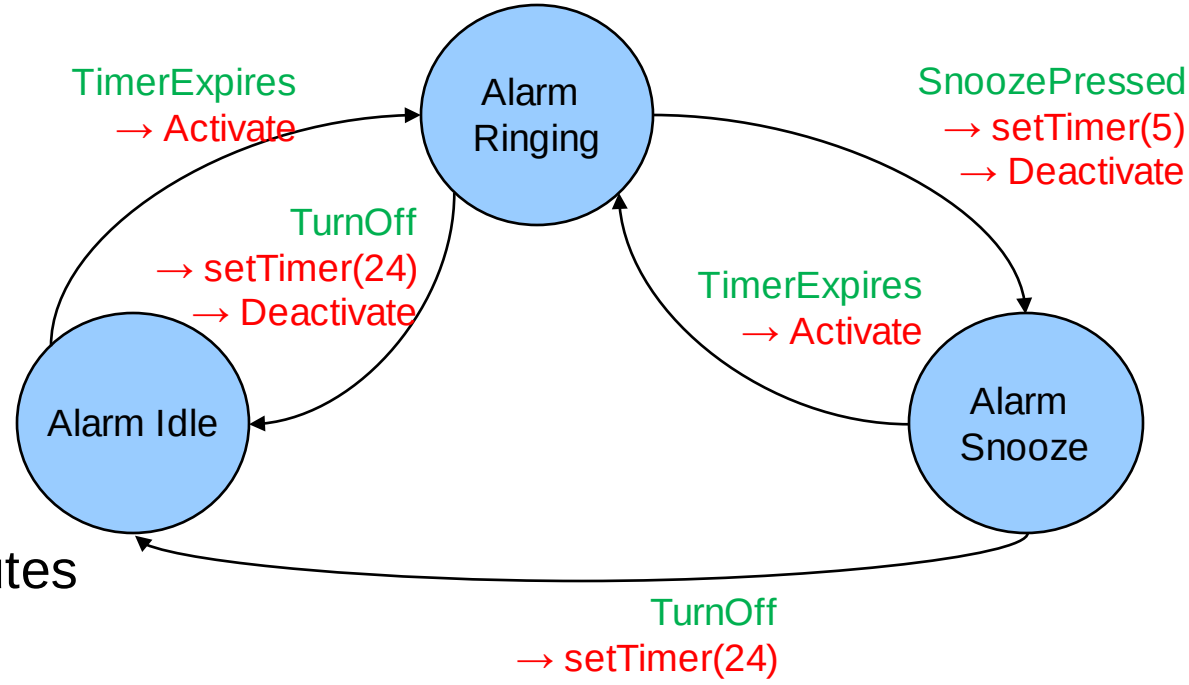


Example: Alarm Clock

- What are possible states?
- What are the inputs?
- What are the actions?
- What control the transitions?

Picture is from:
https://www.instructables.com/

- Sensors:
  - TimerExpires
  - SnoozePressed
  - TrunOff



- Actions:
  - Activate to turn on the alarm
  - setTimer(n) to reset the timer in minutes
  - Deactivate to turn off the alarm