
Engineering Technology (ENGR 101)

Finite State Machine (FSM)

Mohammad Nekooei

Engineering and Computer Science

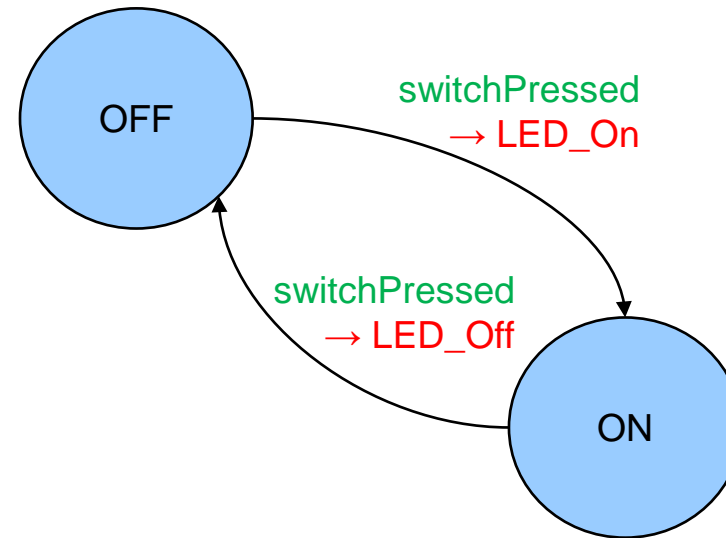
Victoria University of Wellington

Admin

- Assignment 3 has been released
 - Due date is June 1, 19:00 (Xiamen Time)
 - This assignment is individual. You must **not** work in groups.
- Lab 6 has been extended to May 27, 19:00 (Xiamen Time)
- Students who have not submitted their lab projects
 - Assignments 12% of final grade
 - Labs & project 38% of final grade
 - https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/XMUT101CourseOutline

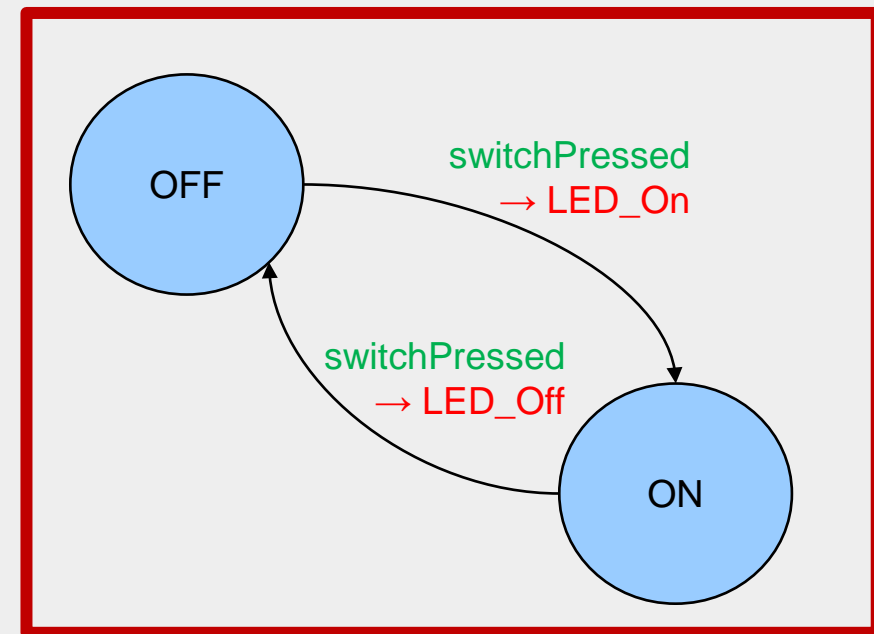
Pushbutton Switch tweaking

- In the previous sketch, while the switch is pressed and off while the switch is not pressed.
- How we can 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 switchPressed = 0;
const int OFF = 0;
const int ON = 1;
int 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 (FMS)

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

- 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*

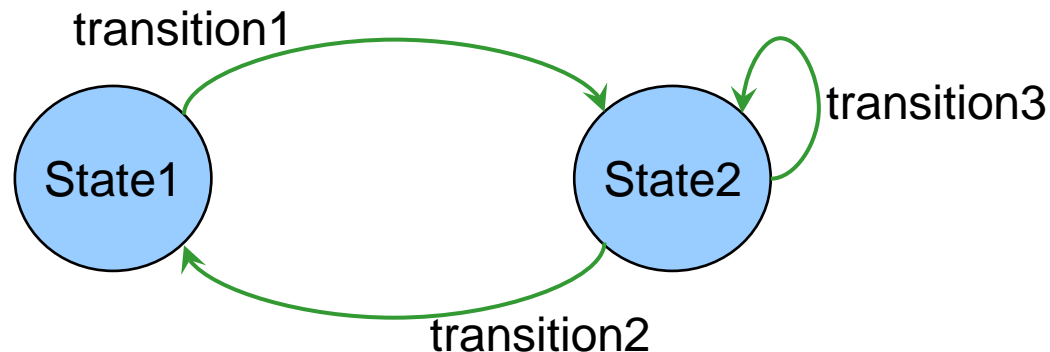
Finite means limited in size. Opposite of Infinite ∞ .

State is the condition or remembered information of a system.

- Typically:

- draw labelled circles for the **states**
- draw labelled arrows for the **transitions** between them

Transition when something changes from one state to another.



Kinds of Finite State Machines.

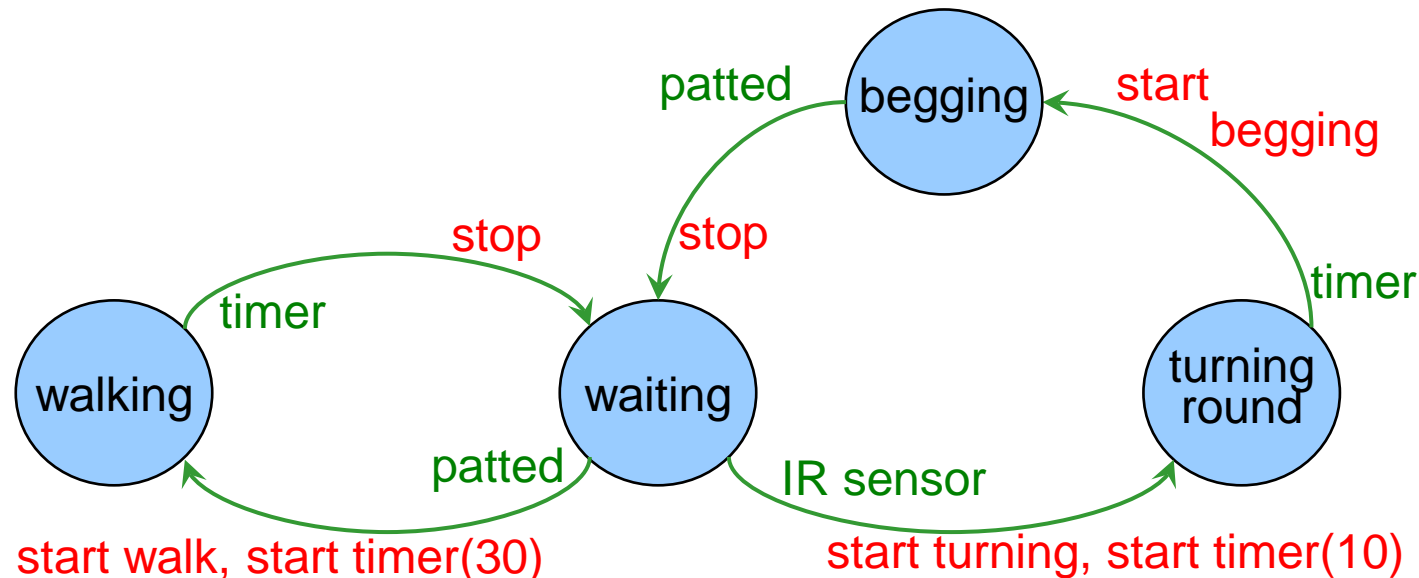
- Describing/specifying an active system
 - **states** the Arduino sketch can be in
 - **input/signals/sensors** that the Arduino may respond to
 - **actions** that the Arduino will perform

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 7)

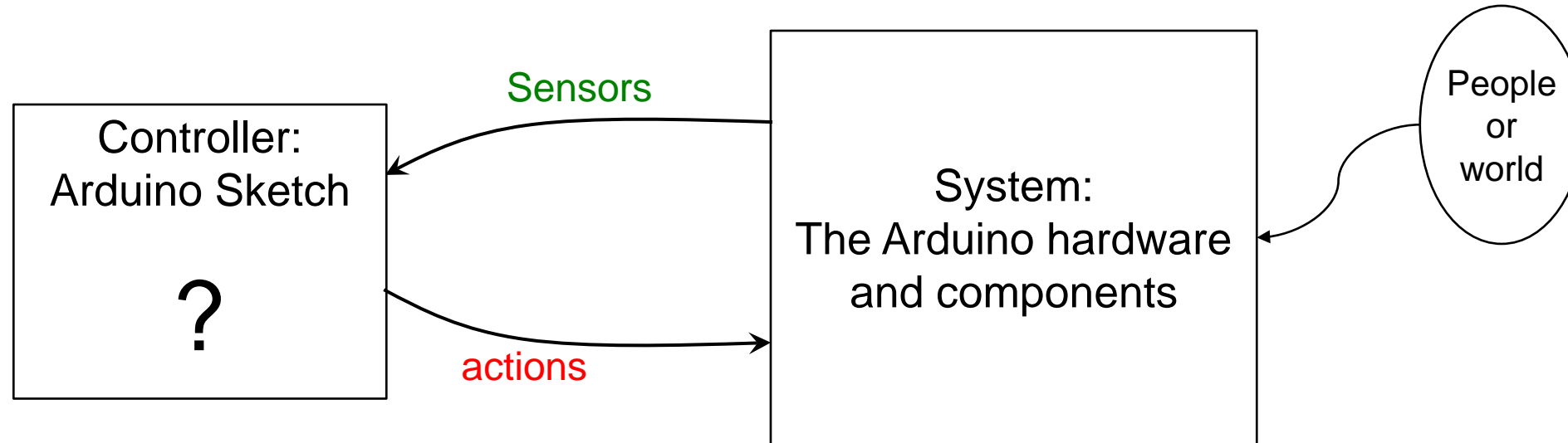
...



Auti designed by
Helen Andreae

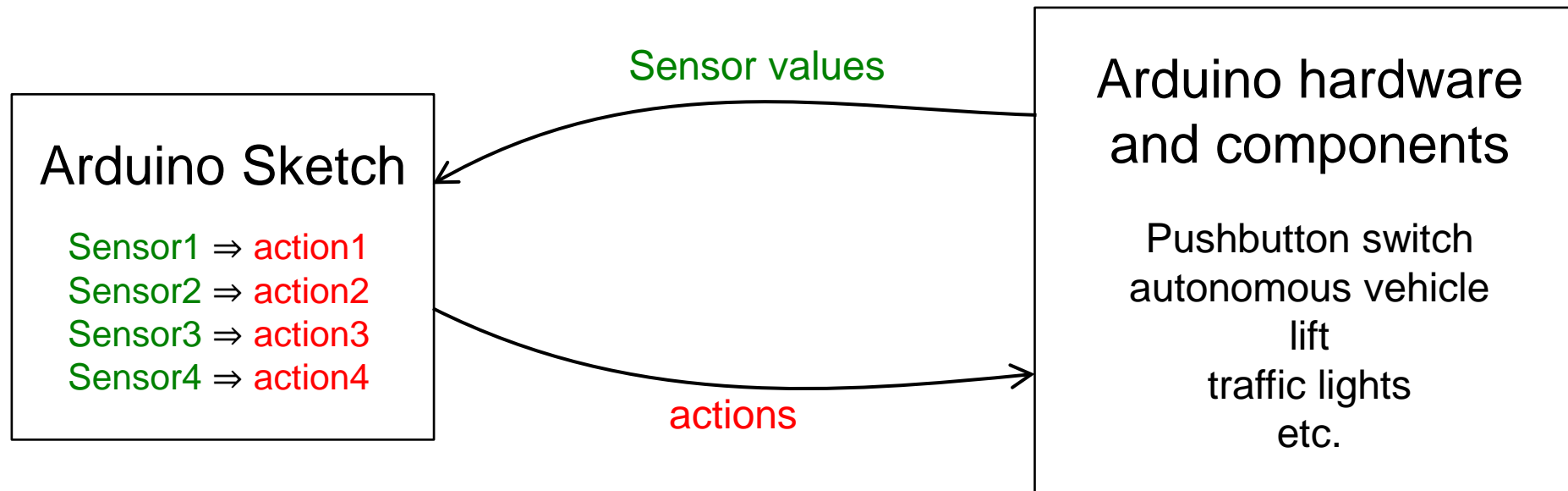
Controllers - Arduino

- 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

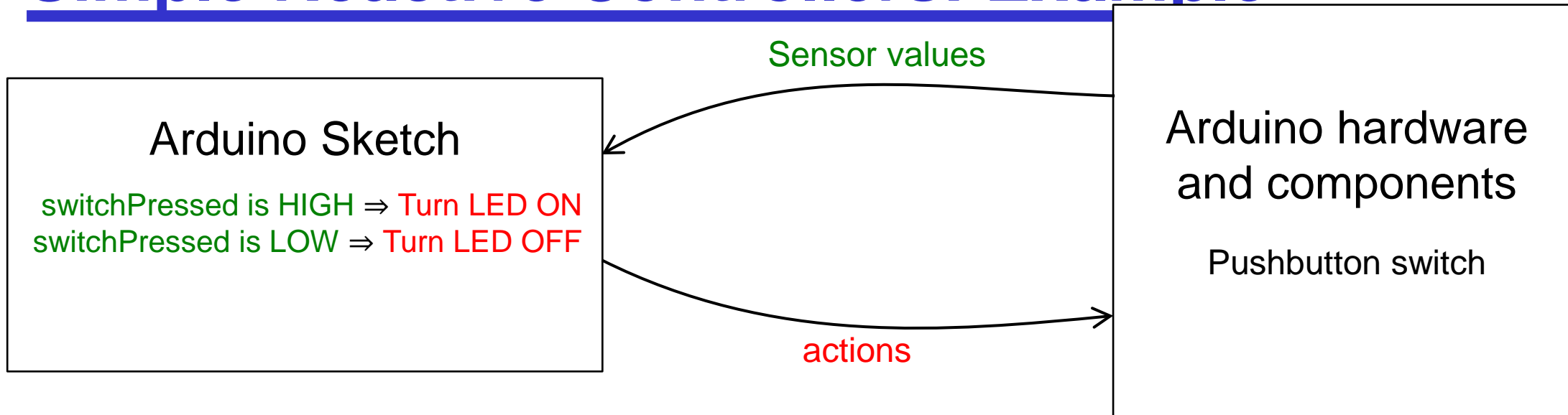


Simple Reactive Controllers.

- 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



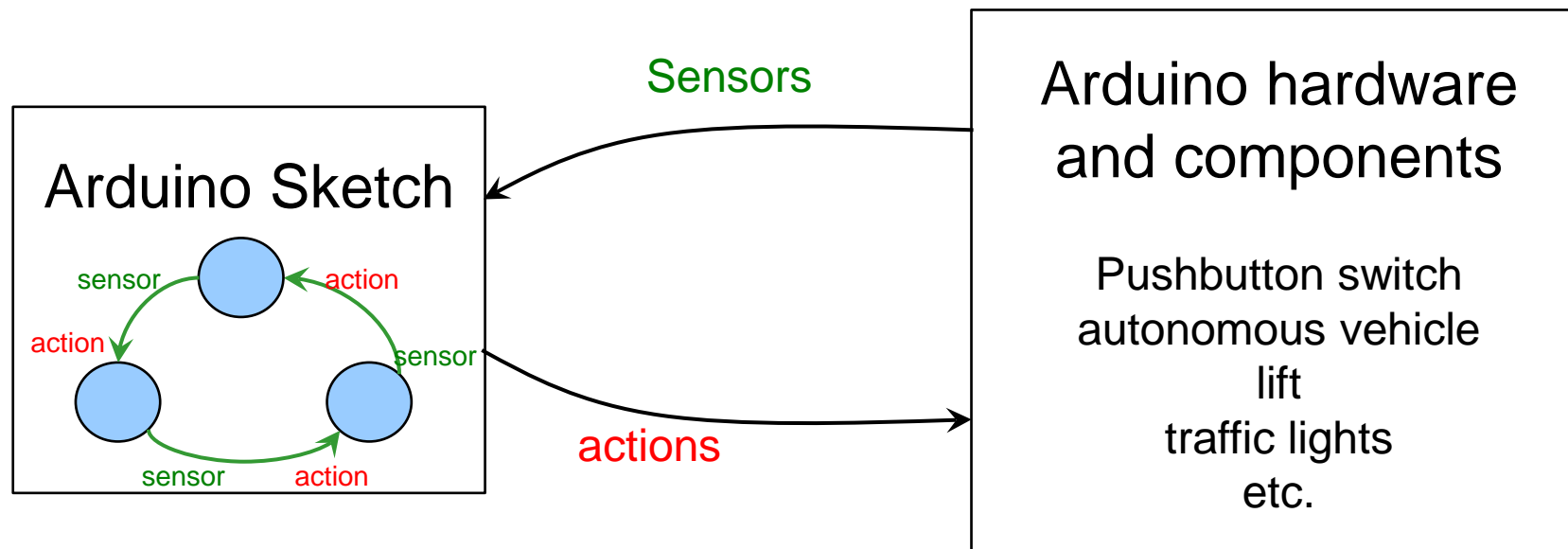
Simple Reactive Controllers: Example



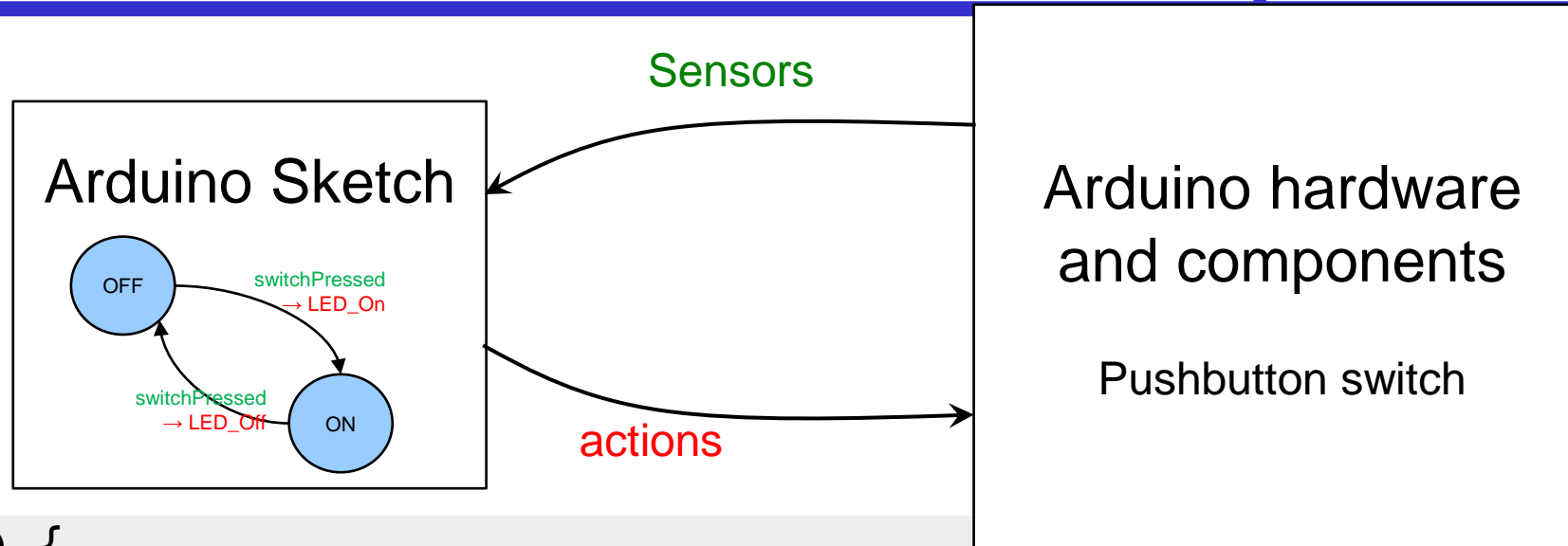
```
void loop() {  
  // Get the input  
  switchPressed = digitalRead(switchPin);  
  // Change the output  
  if (switchPressed == HIGH ) {  
    digitalWrite(LEDpin, HIGH);  
  }  
  else if (switchPressed == LOW ) {  
    digitalWrite(LEDpin, LOW);  
  }  
}
```

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.



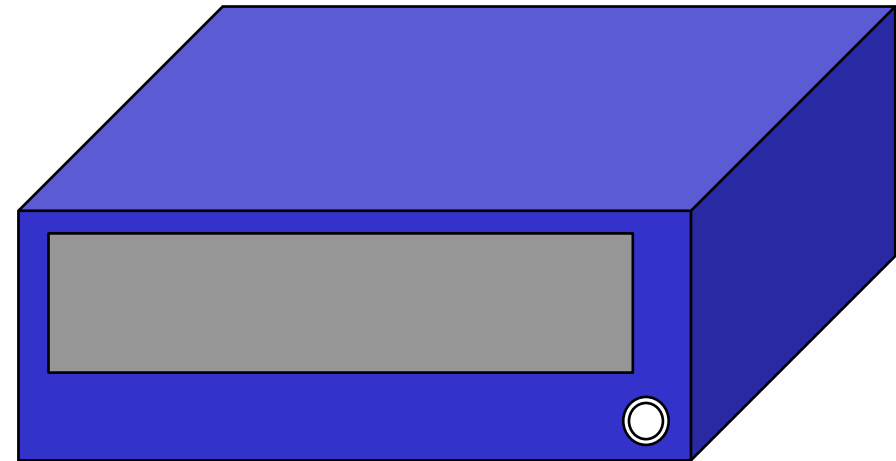
Simple Reactive Controllers: Example



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

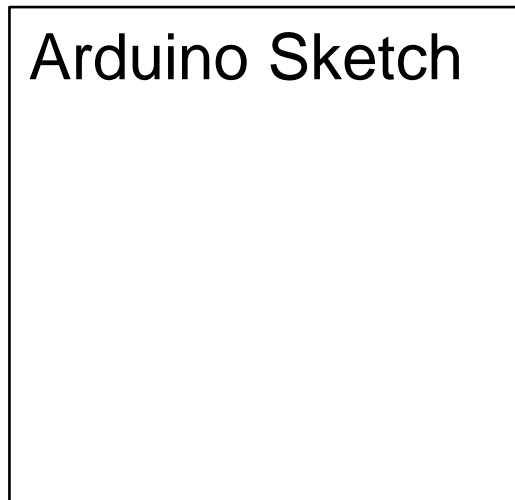
Example: Document Holder with Arduino

- 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



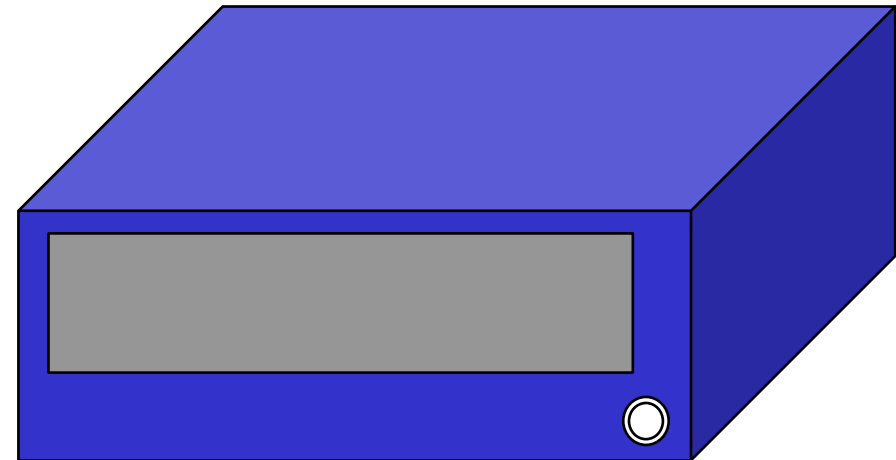
Example: Document Holder

- 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



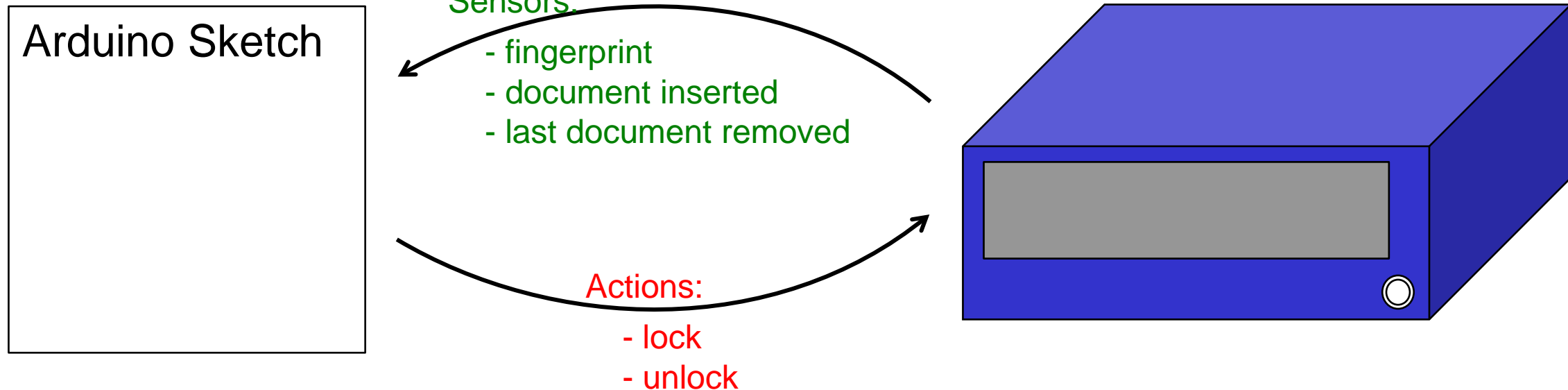
Sensors:

- fingerprint
- document inserted
- last document removed



Example: Electronic Document Holder

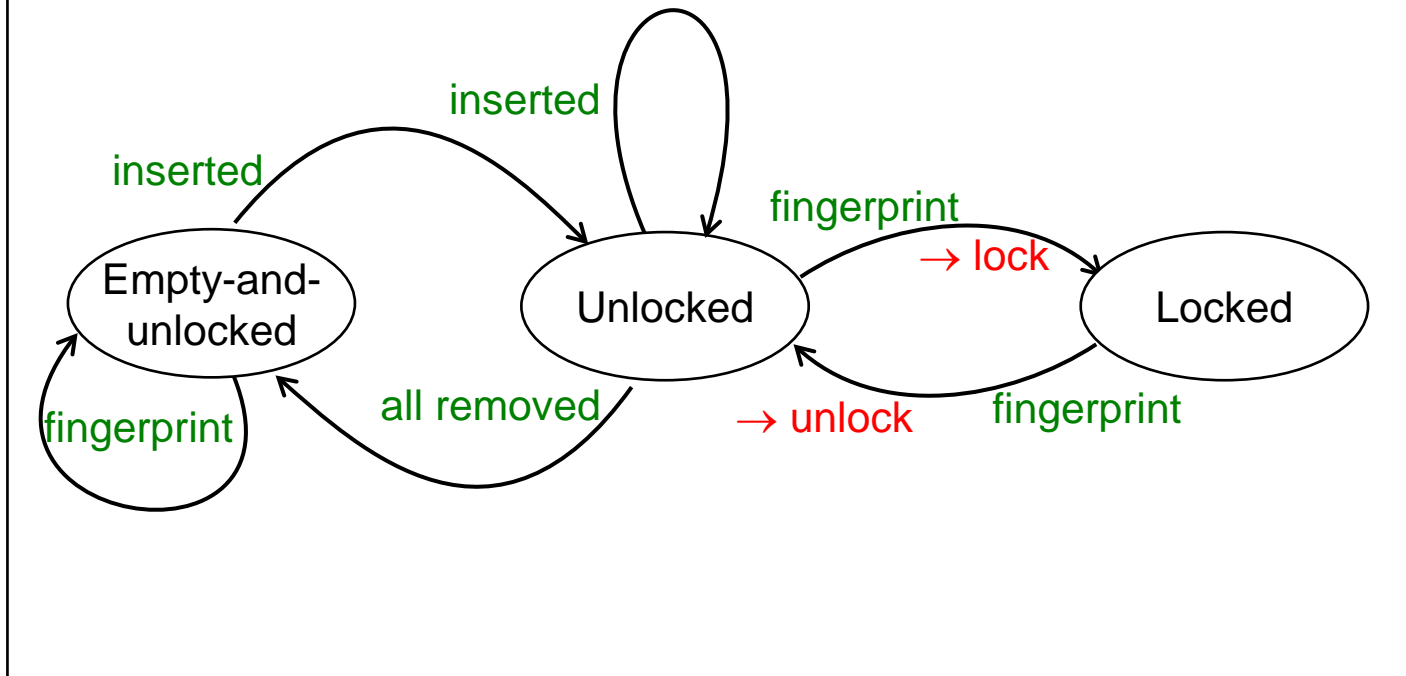
- Document holder with a fingerprint lock, and a document sensor
 - It has to be unlocked to put documents in or out.
 - If there are no documents in it, it stays unlocked.
 - If there are documents in it, then the fingerprint lock will lock/unlock the holder



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.

Arduino Sketch

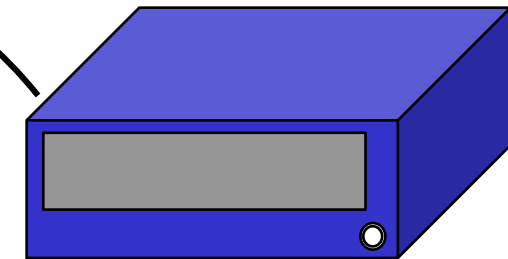


Sensors:

- fingerprint
- inserted
- all removed

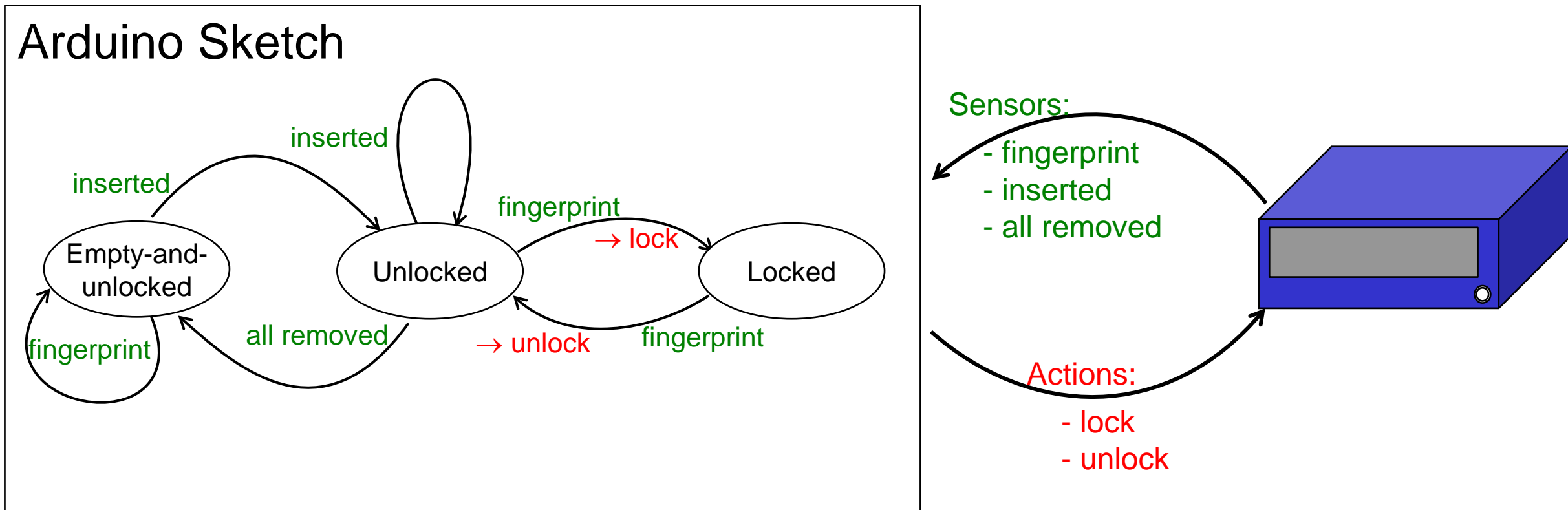
Actions:

- lock
- unlock



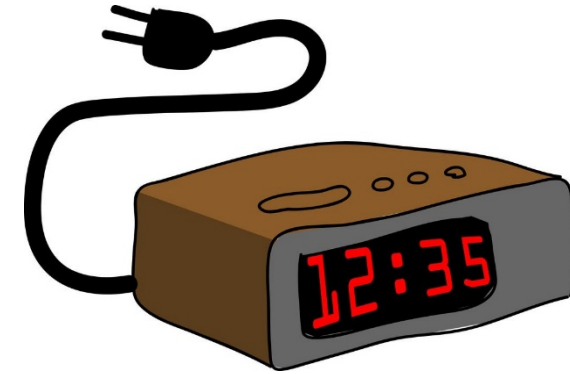
Example: Electronic Document Holder

- 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
- TurnOff

- Actions:

- **Activate** to turn on the alarm
- **resetTimer** to reset the timer
- **Deactivate** to turn off the alarm

