ESP32 Interrupts

Chen-Hanson Ting

SVFIG

January 25, 2020

ESP32 Interrupts

- AI bot needs interrupt control to handle events at the instinct level.
- ESP32eforth has to be extended to handle real time interrupts.
- A light box experiment is used to demonstrate interrupt control.

Timer Interrupt Example

```
volatile int interruptCounter;
int totalInterruptCounter;
hw_timer_t * timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
void IRAM_ATTR onTimer() {
  portENTER_CRITICAL_ISR(&timerMux);
  interruptCounter++;
  portEXIT_CRITICAL_ISR(&timerMux);}
void setup() {
  timer = timerBegin(0, 80, true);
  timerAttachInterrupt(timer, &onTimer, true);
  timerAlarmWrite(timer, 1000000, true);
  timerAlarmEnable(timer);}
void loop() {
  if (interruptCounter > 0) {
    portENTER_CRITICAL(&timerMux);
    interruptCounter--;
    portEXIT_CRITICAL(&timerMux);
    totalInterruptCounter++;
    Serial.print("An interrupt as occurred. Total number: ");
    Serial.println(totalInterruptCounter);}}
```

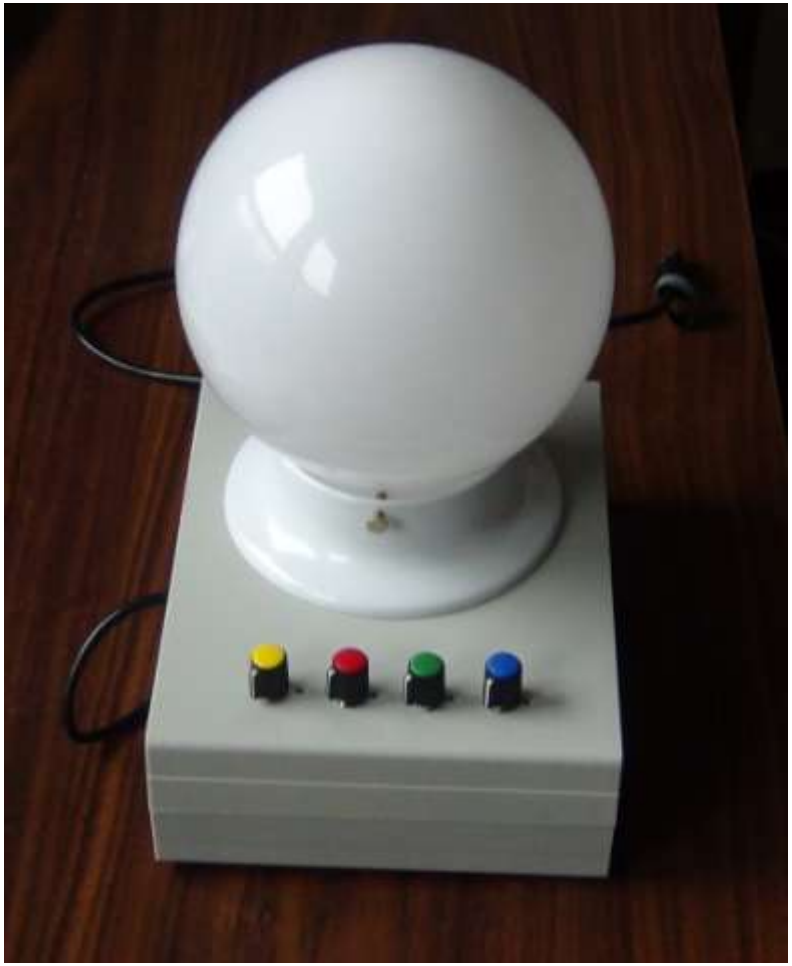
Tricolor Lightbox

- One white LED, one RGB tricolor LED.
- 4 potentiometers to control brightness of LEDs
- Interrupts generated by an internal timer every 100 ms.

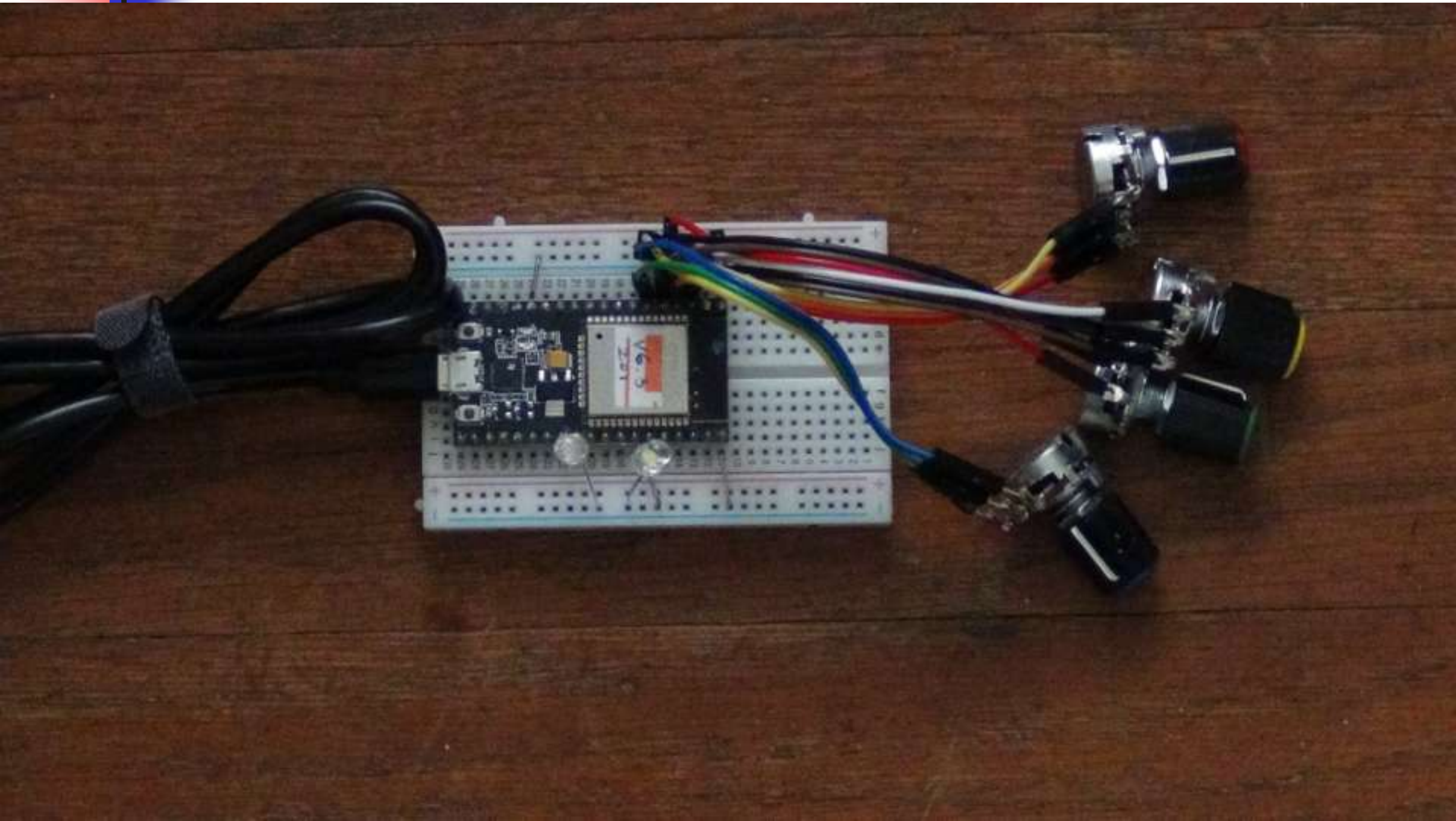
5 Timers

- Timer 0 is set up to generate interrupts every 100 ms.
- 4 other timers generate PWM waves to driver 4 LEDs.

Tricolor Lightbox



Tricolor Lightbox Circuit



Interrupt Routine

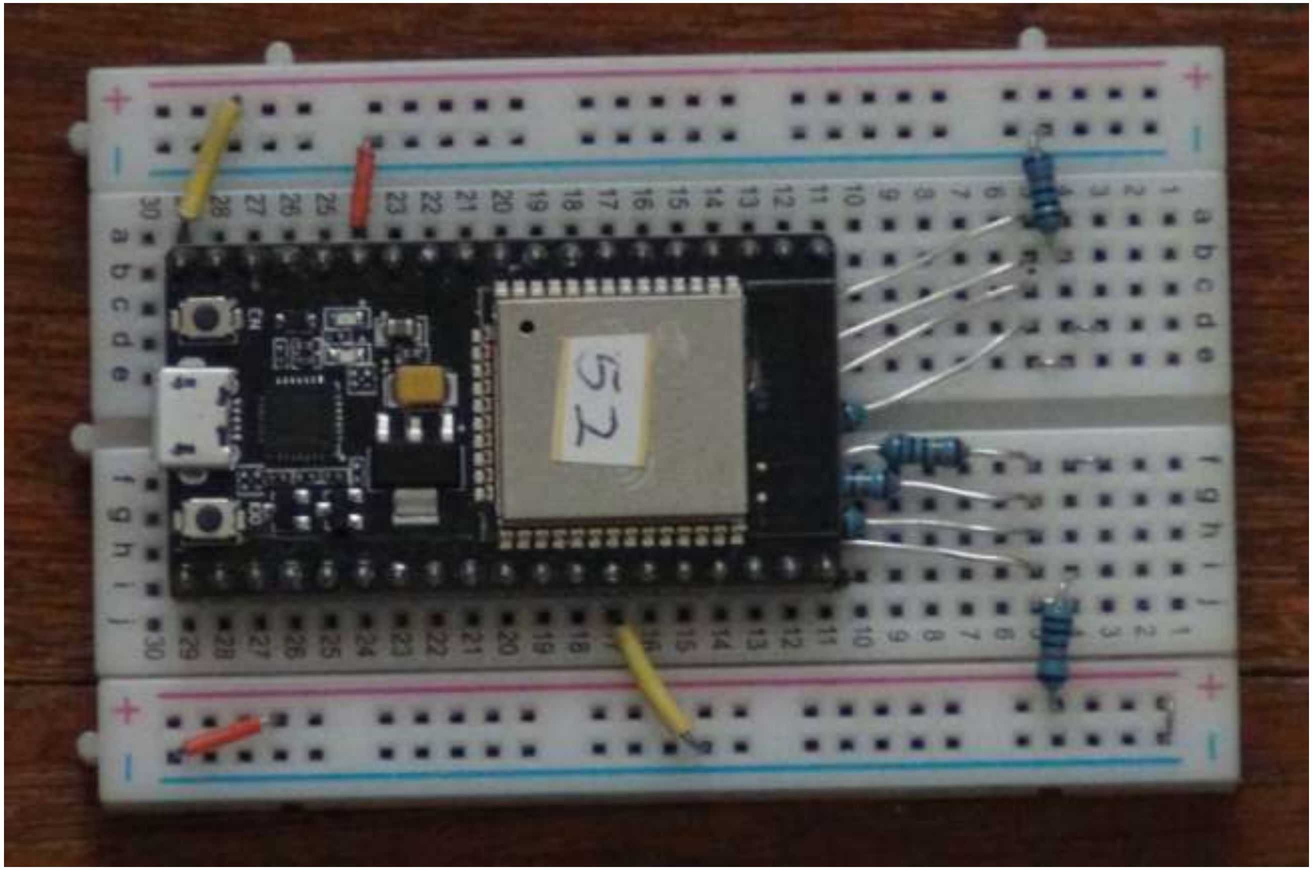
```
hw_timer_t * timer = NULL;
portMUX_TYPE timerMux =
    portMUX_INITIALIZER_UNLOCKED;
void IRAM_ATTR onTimer() {
  portENTER_CRITICAL_ISR(&timerMux);
  interruptCounter++;
  white_value=analogRead(white_in);
  ledcWrite(white_channel,white_value);
  red_value=analogRead(red_in);
  ledcWrite(red_channel,red_value);
  green_value=analogRead(green_in);
  ledcWrite(green_channel,green_value);
  blue_value=analogRead(blue_in);
  ledcWrite(blue_channel,blue_value);
  portEXIT_CRITICAL_ISR(&timerMux);
}
```

Interrupt Setup

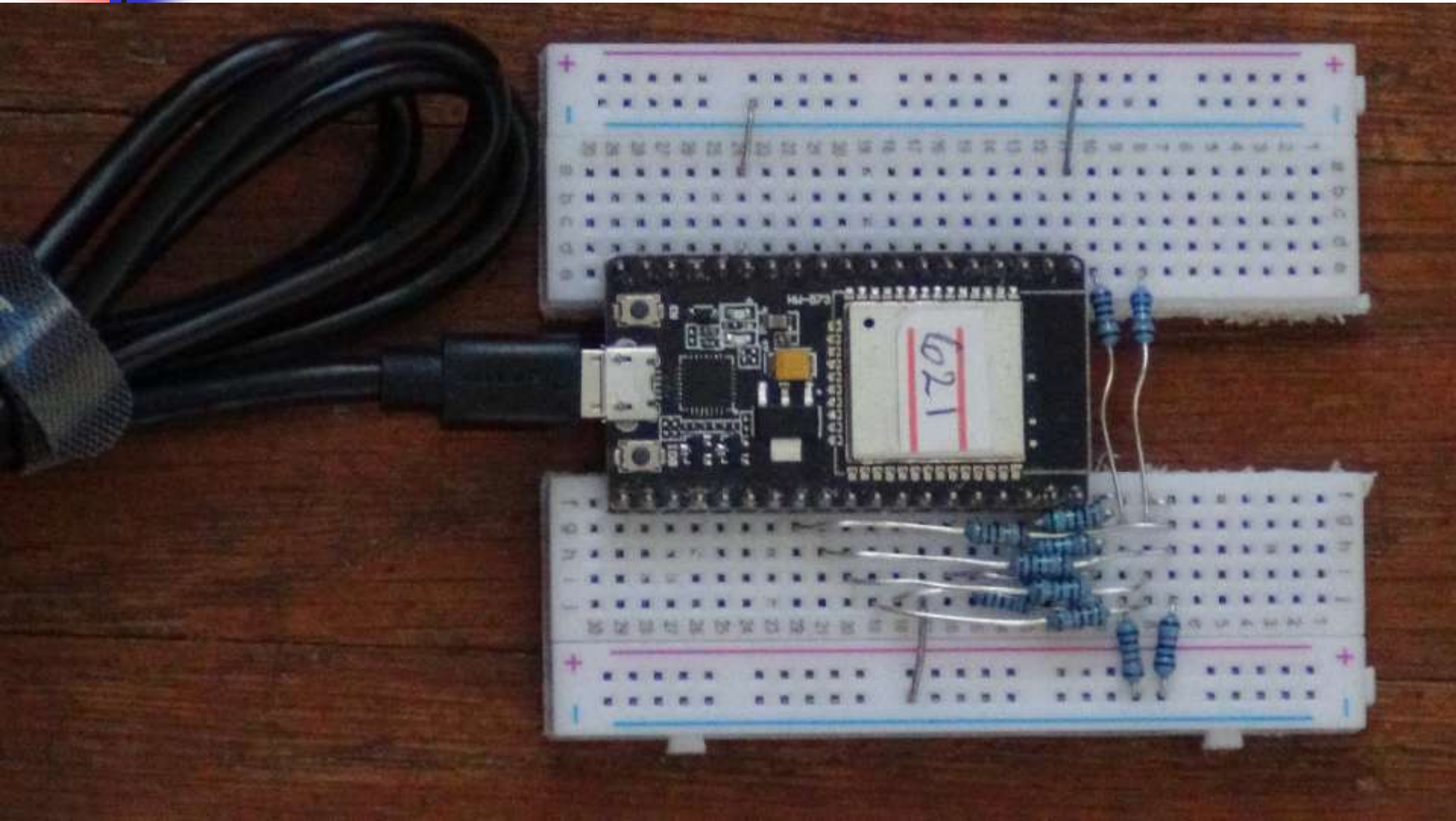
```
timer = timerBegin(0, 80, true);
timerAttachInterrupt(timer, &onTimer, true);
timerAlarmWrite(timer, 100000, true);
timerAlarmEnable(timer);
ledcSetup(white_channel, frequency, resolution);
ledcAttachPin(white_out, white_channel);
ledcWrite(white_channel, dutyCycle);
ledcSetup(red_channel, frequency, resolution);
ledcAttachPin(red_out, red_channel);
ledcWrite(red_channel, dutyCycle);
ledcSetup(green_channel, frequency, resolution);
ledcAttachPin(green_out, green_channel);
ledcWrite(green_channel, dutyCycle);
ledcSetup(blue_channel, frequency, resolution);
ledcAttachPin(blue_out, blue_channel);
ledcWrite(blue_channel, dutyCycle);
```

Demo

ESP32 Organ Kit



New ESP32 Kit



Questions?

Thank You!