Laporan Tugas #1 Lampu Toggle

EL3014 SiStem Mikroprosesor

Nama:

NIM

Contents

[1 Sampling Dengan Loop 1](#_Toc63761046)

[1.1 Flowchart 2](#_Toc63761047)

[1.2 Source Code 2](#_Toc63761048)

[1.3 Penjelasan 2](#_Toc63761049)

[1.4 Hasil simulasi 2](#_Toc63761050)

[2 Sampling Dengan Delay 3](#_Toc63761051)

[2.1 Flowchart 3](#_Toc63761052)

[2.2 Source Code 3](#_Toc63761053)

[2.3 Penjelasan 3](#_Toc63761054)

[2.4 Hasil simulasi 3](#_Toc63761055)

[3 Sampling Dengan Interupsi Timer 4](#_Toc63761056)

[3.1 Flowchart 4](#_Toc63761057)

[3.2 Source Code 4](#_Toc63761058)

[3.3 Penjelasan 4](#_Toc63761059)

[3.4 Hasil simulasi 4](#_Toc63761060)

[4 Wait Loop 5](#_Toc63761061)

[4.1 Flowchart 5](#_Toc63761062)

[4.2 Source Code 5](#_Toc63761063)

[4.3 Penjelasan 5](#_Toc63761064)

[4.4 Hasil simulasi 5](#_Toc63761065)

[5 Interupsi Eksternal 6](#_Toc63761066)

[5.1 Flowchart 6](#_Toc63761067)

[5.2 Source Code 6](#_Toc63761068)

[5.3 Penjelasan 6](#_Toc63761069)

[5.4 Hasil simulasi 6](#_Toc63761070)

[6 Referensi 6](#_Toc63761071)

Daftar Gambar

[Gambar 1 Skema rangkaian 3](#_Toc63761072)

[Gambar 2 Contoh Flowchart 3](#_Toc63761073)

[Gambar 2 Contoh Flowchart 4](#_Toc63761074)

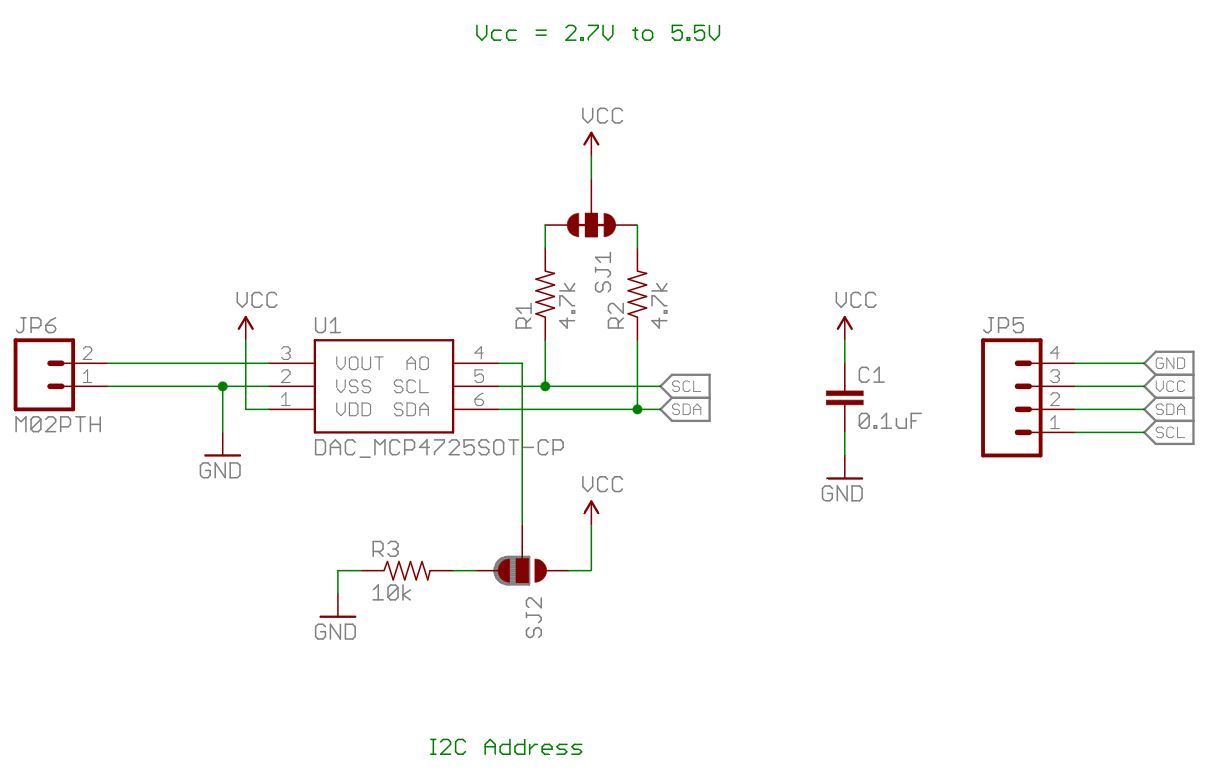
[Gambar 2 Contoh Flowchart 5](#_Toc63761075)

[Gambar 2 Contoh Flowchart 6](#_Toc63761076)

[Gambar 2 Contoh Flowchart 7](#_Toc63761077)

# Sampling Dengan Loop

Skema rangkaian dicantumkan cukup sekali jika sama untuk semuanya.



Gambar 1 Skema rangkaian

## Flowchart



Gambar 2 Contoh Flowchart

## Source Code

|  |
| --- |
| #include <Arduino.h>  #define PIN\_LED PIN\_A5  #define PIN\_BUTTON 3  #define LOOP\_PERIOD\_IN\_MS 500  bool lamp\_state = 0;  bool last\_button\_state;  void setup() {    // put your setup code here, to run once:    Serial.begin(115200);    pinMode(PIN\_LED, OUTPUT);    pinMode(PIN\_BUTTON, INPUT);    digitalWrite(PIN\_LED, lamp\_state);    last\_button\_state = digitalRead(PIN\_BUTTON);  }  void loop() {    // put your main code here, to run repeatedly:    bool curr\_button\_state = digitalRead(PIN\_BUTTON);    if (curr\_button\_state != last\_button\_state && last\_button\_state == 0){      lamp\_state = !lamp\_state;      digitalWrite(PIN\_LED, lamp\_state);    }    last\_button\_state = curr\_button\_state;  } |

## Penjelasan

Sistem ini bekerja dengan cara sampling nilai button. Jika button telah berubah state dari nilai sebelumnya, dan terjadi keadaan rising (dari 0 hingga 1), maka lampu akan berubah state.

Masalah dari sistem ini adalah akan terjadi debouncing. Pengambilan sampling dilakukan dengan sangat cepat sekali. Saat tombol ditekan, nilai yang terjadi akan bolak balik antara 0 dan 1, sehingga lampu akan berkedip dengan cepat. Karena nilai sampling sangat cepat sekali, maka setiap perubahan dari tombol akan terdeteksi oleh sistem, dan lampu akan berkedip dengan cepat. Apalagi penulis menggunakan kabel dan tidak menggunakan switch komponen.

## Hasil simulasi

Hasil simulasi diuraikan di sini. Dapat berupa video, ataupun penjelasan hasil simulasi

# Sampling Dengan Delay

## Flowchart



Gambar 3 Contoh Flowchart

## Source Code

|  |
| --- |
| #include <Arduino.h>  #define PIN\_LED PIN\_A5  #define PIN\_BUTTON 3  #define LOOP\_PERIOD\_IN\_MS 500  bool lamp\_state = 0;  bool last\_button\_state;  void setup() {    // put your setup code here, to run once:    Serial.begin(115200);    pinMode(PIN\_LED, OUTPUT);    pinMode(PIN\_BUTTON, INPUT);    digitalWrite(PIN\_LED, lamp\_state);    last\_button\_state = digitalRead(PIN\_BUTTON);  }  void loop() {    // put your main code here, to run repeatedly:    bool curr\_button\_state = digitalRead(PIN\_BUTTON);    if (curr\_button\_state != last\_button\_state && last\_button\_state == 0){      lamp\_state = !lamp\_state;      digitalWrite(PIN\_LED, lamp\_state);    }    last\_button\_state = curr\_button\_state;    delay(LOOP\_PERIOD\_IN\_MS);  } |

## Penjelasan

Sistem ini bekerja dengan cara yang sama dengan sistem sebelumnya. Sistem ini bekerja dengan cara sampling nilai button. Jika button telah berubah state dari nilai sebelumnya, dan terjadi keadaan rising (dari 0 hingga 1), maka lampu akan berubah state. Lalu, loop akan didelay selama waktu tertentu (500 ms dalam sistem yang dibuat penulis).

Dengan menggunakan sistem ini, debouncing lebih tidak terdeteksi, karena pada saat kondisi rising pertama, maka sistem tidak akan mendeteksi hingga selesai waktu delay.

## Hasil simulasi

Hasil simulasi diuraikan di sini. Dapat berupa video, ataupun penjelasan hasil simulasi

# Sampling Dengan Interupsi Timer

## Flowchart



Gambar 4 Contoh Flowchart

## Source Code

|  |
| --- |
| #include <Arduino.h>  #include "TimerOne.h"  #define PIN\_LED PIN\_A5  #define PIN\_BUTTON 3  #define LOOP\_PERIOD\_IN\_US 500000  void toggleLED();  bool lamp\_state = 0;  bool last\_button\_state;  void setup() {    // put your setup code here, to run once:    Serial.begin(115200);    pinMode(PIN\_LED, OUTPUT);    pinMode(PIN\_BUTTON, INPUT);    digitalWrite(PIN\_LED, lamp\_state);    last\_button\_state = digitalRead(PIN\_BUTTON);      Timer1.initialize(LOOP\_PERIOD\_IN\_US);    Timer1.attachInterrupt(toggleLED);  }    void loop() {    // put your main code here, to run repeatedly:  }  void toggleLED(){    bool curr\_button\_state = digitalRead(PIN\_BUTTON);    if (curr\_button\_state != last\_button\_state && last\_button\_state == 0){      lamp\_state = !lamp\_state;      digitalWrite(PIN\_LED, lamp\_state);    }    last\_button\_state = curr\_button\_state;  } |

## Penjelasan

Sistem ini bekerja dengan cara membuat fungsi toggleLED pada waktu. Sistem akan mengamati kondisi button pada selang waktu yang konstan. Pengamatan tersebut menggunakan timer interrupt. Setelah itu, jika button state berubah dan sebelumnya button 0 (button ditekan, rising), maka LED akan berubah state.

Interrupt timer memliki keuntungan pada bagian rate fungsi dijalankan yang konstan. Syarat rate konstan adalah fungsi tidak boleh dijalankan lebih lama dari periode interrupt.

## Hasil simulasi

Hasil simulasi diuraikan di sini. Dapat berupa video, ataupun penjelasan hasil simulasi

# Wait Loop

## Flowchart



Gambar 5 Contoh Flowchart

## Source Code

|  |
| --- |
| #include <Arduino.h>  #define PIN\_LED PIN\_A5  #define PIN\_BUTTON 3  #define LOOP\_PERIOD\_IN\_MS 500  bool lamp\_state = 0;  bool curr\_button\_state;  void setup() {  // put your setup code here, to run once:  Serial.begin(115200);  pinMode(PIN\_LED, OUTPUT);  pinMode(PIN\_BUTTON, INPUT);  digitalWrite(PIN\_LED, lamp\_state);  curr\_button\_state = digitalRead(PIN\_BUTTON);  }  void loop() {  // put your main code here, to run repeatedly:  do{  curr\_button\_state = digitalRead(PIN\_BUTTON);  }while (curr\_button\_state != 0);  do{  curr\_button\_state = digitalRead(PIN\_BUTTON);  }while (curr\_button\_state != 1);  // proses  lamp\_state = !lamp\_state;  digitalWrite(PIN\_LED, lamp\_state);  } |

## Penjelasan

Sistem bekerja dengan cara menunggu sistem hingga terjadi perubahan. Dapat dilihat bahwa dari do\_while yang pertama, nilai curr\_button\_state pasti bernilai 0. (sistem akan menunggu hingga curr\_button\_state bernilai 0). Kemudian, sistem akan menunggu kembali apakah button berubah menjadi 1. Hal ini berarti terjadi rising button. Lampu akan ditoggle, dan sistem ini diulang kembali.

Sistem ini juga rawan debouncing, karena waktu pengecekan sangat cepat (tidak ada delay).

## Hasil simulasi

Hasil simulasi diuraikan di sini. Dapat berupa video, ataupun penjelasan hasil simulasi

# Interupsi Eksternal

## Flowchart



Gambar 6 Contoh Flowchart

## Source Code

|  |
| --- |
| #include <Arduino.h>  #define PIN\_LED PIN\_A5  #define PIN\_BUTTON 3  #define LOOP\_PERIOD\_IN\_US 500000  void toggleLED();  bool lamp\_state = 0;  bool last\_button\_state;  void setup() {    // put your setup code here, to run once:    Serial.begin(115200);    pinMode(PIN\_LED, OUTPUT);    digitalWrite(PIN\_LED, lamp\_state);      attachInterrupt(digitalPinToInterrupt(PIN\_BUTTON), toggleLED, RISING);  }    void loop() {    // put your main code here, to run repeatedly:    digitalWrite(PIN\_LED, lamp\_state);  }  void toggleLED(){    lamp\_state = !lamp\_state;    } |

## Penjelasan

`

Sistem ini bekerja dengan cara membuat pin button menjadi interrupt. Pada arduino uno, interrupt pin terdapat pada pin 2 dan pin 3. Sistem akan mendeteksi apakah terjadi kondisi rising pada button. Jika terjadi, toggleLED akan dijalankan, dan mengubah state dari lampu. LED akan berubah statenya yang dilakukan pada loop. Alasan menggunakan LED pada loop agar callback interrupt memiliki waktu jalan fungsi sekecil mungkin.

Sistem ini memiliki keuntungan pada kita tidak perlu mengecek pin secara berkala. Sistem akan dikabari jika terjadi rising. Callback hanya dijalankan sekali sekali saat terjadi rising. Sistem akan memiliki konsumsi daya yang lebih hemat.

## Hasil simulasi

Hasil simulasi diuraikan di sini. Dapat berupa video, ataupun penjelasan hasil simulasi

# Referensi

Referensi yang dipakai