Modul 1: Digital Input / Output

**Farrel Dzaudan Naufal 13318048, Kelompok: 18**

Tanggal: 31 Januari 2020, Assisten: Krisna Diastama 13317016

TF 2207 - Laboratorium Teknik Fisika II

# Tujuan Percobaan

1. Menentukan tingkat kecerahan LED
2. Menentukan alur kerja program pencacah
3. Menentukan nilai display periode yang optimal
4. Menentukan alur kerja program stopwatch
5. Menentukan tabel translasi/penerjemahan output keypad
6. Menentukan alur kerja program kalkulator

# Alat dan Bahan

Table 2.1 Nama Tabel (gunakan References->Insert Caption)

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Banyak | Keterangan |
| 1 | Laptop | 1 | Telah terpasang software Arduino |
| 2 | Kit praktikum | 1 | Arduino dan komponen-komponen elektrik (7-segment, keypad, IC decoder) yang sudah terpasang pada PCB |

# Teori Dasar

Arduino Uno, microcontroller 8-bit ATmega328P, dapat menerima input atau mengeluarkan output dengan jumlah sebanyak 20 input/output. Input dan output ini diterima dan dikeluarkan melalui pin arduino yang terbagi menjadi 2 kategori: pin digital dan pin analog. Pin digital yang berjumlah 14 pin mempunyai beberapa fungsi khusus untuk pin spesifik, yaitu pin serial Rx (D0) dan Tx (D1) untuk menangani data serial TTL, pin interupsi eksternal (D2 dan D3) untuk membuat trigger interupsi berdasarkan suatu input, pin PWM (D3,D5,D6,D9, dan D11) yang dapat mengeluarkan sinyal Pulse Width Modulation 8-bit, pin SPI SS (D10), MOSI (D11), MISO (D12), dan SCK (D13) yang berfungsi untuk komunikasi antara microcontroller, dan pin LED built-in (D13) yang memiliki LED terhubung pada pin tersebut. Sementara, 6 pin yang tersisa adalah pin Analog yang dapat membaca dan mengeluarkan sinyal dengan volt spesifik dalam range 0 – 5 V dengan resolusi 10-bit[[1]](#footnote-1).

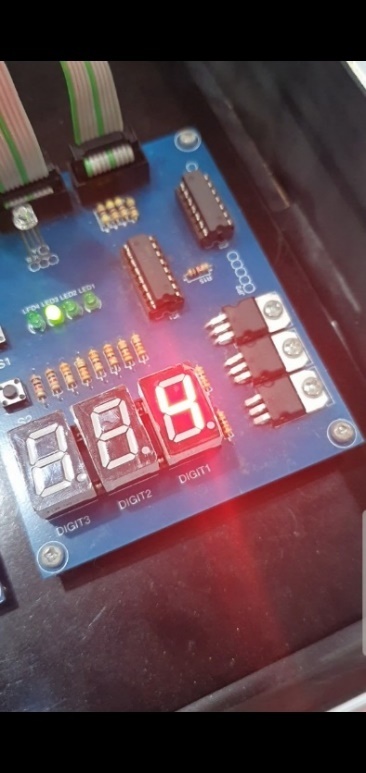
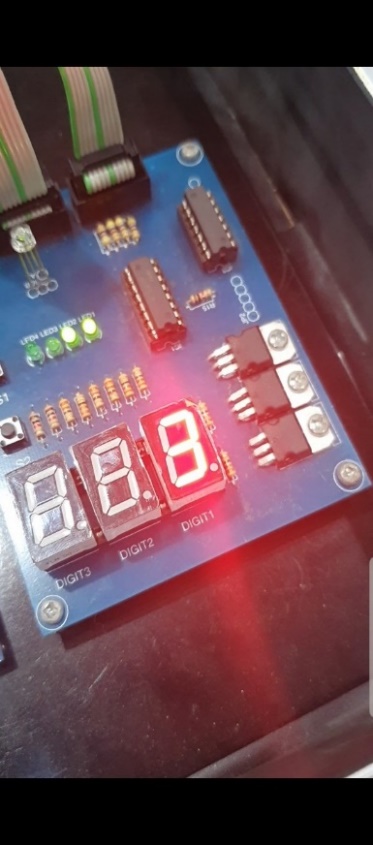
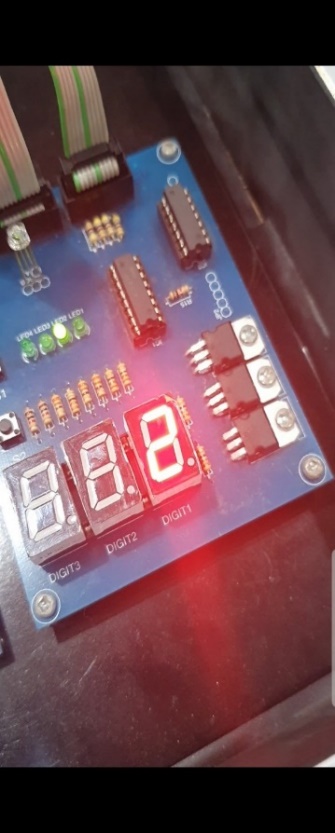
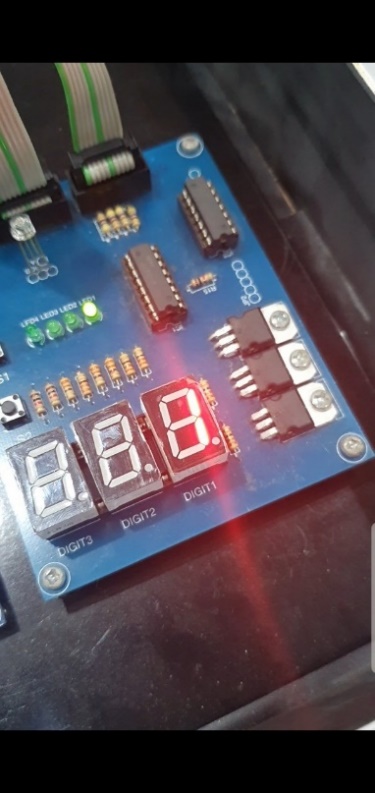
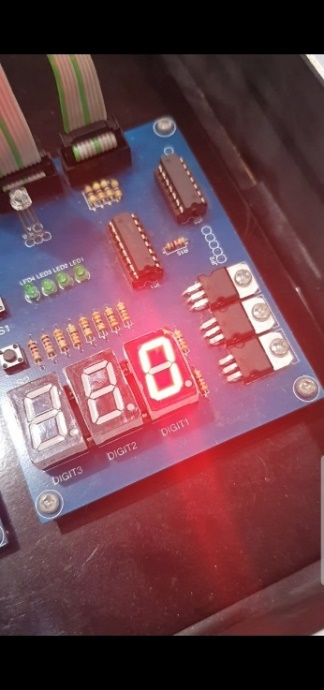
Decoder adalah sebuah Combinational Circuit yang memilih salah satu dari 2n output berdasarkan n input. Namun, sebuah decoder dapat memiliki output lebih sedikit dari 2n. Contohnya, BCD (binary coded decimal) to seven-segment decoder memiliki 4 input dan 7 output, bukan 24 = 16 output. Sementara itu, encoder adalah Combinational Circuit yang melakukan operasi berkebalikan dari decoder. Sebuah encoder memiliki maksimum 2n input dan n output[[2]](#footnote-2).

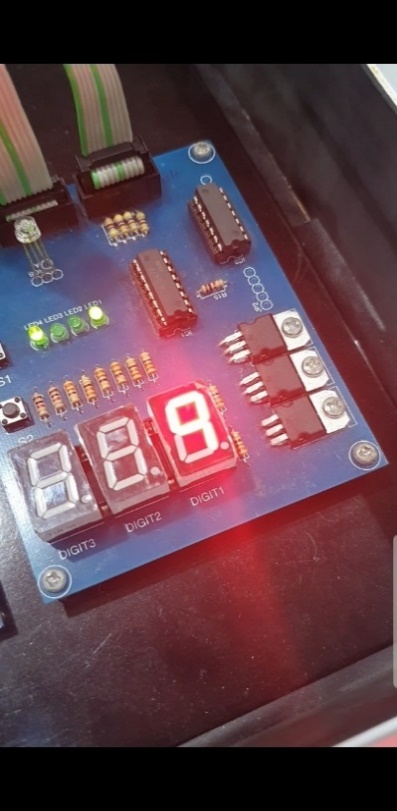
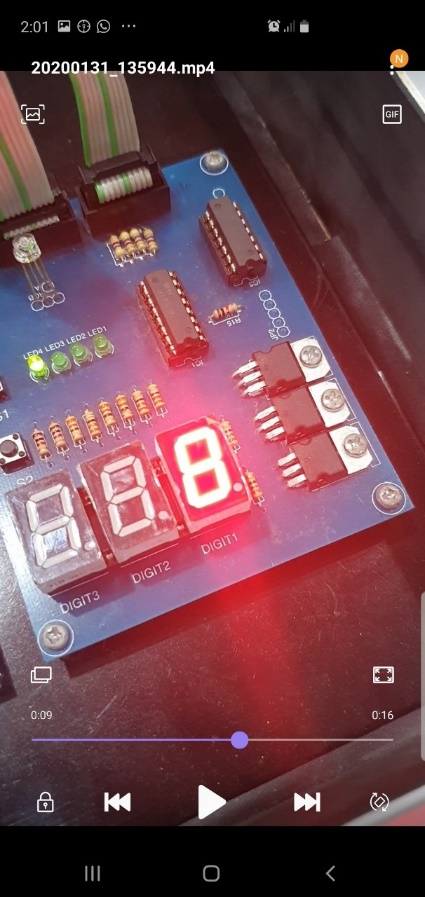
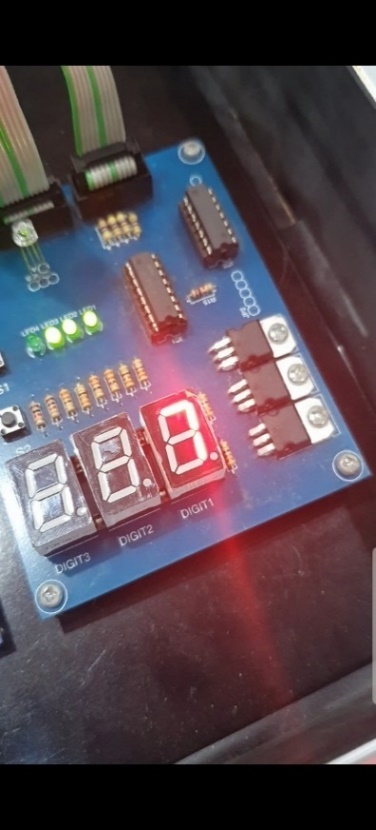
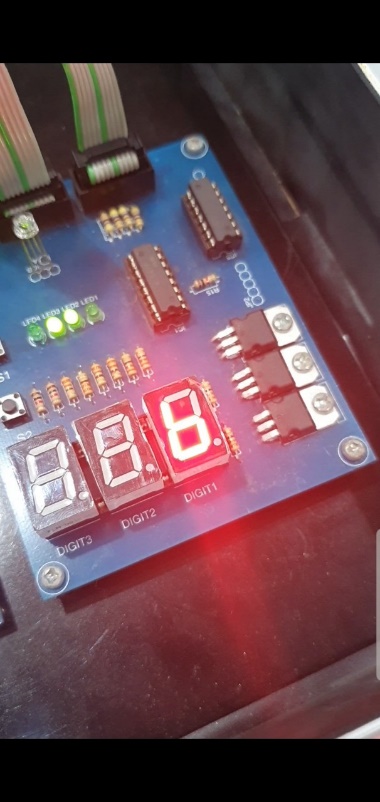
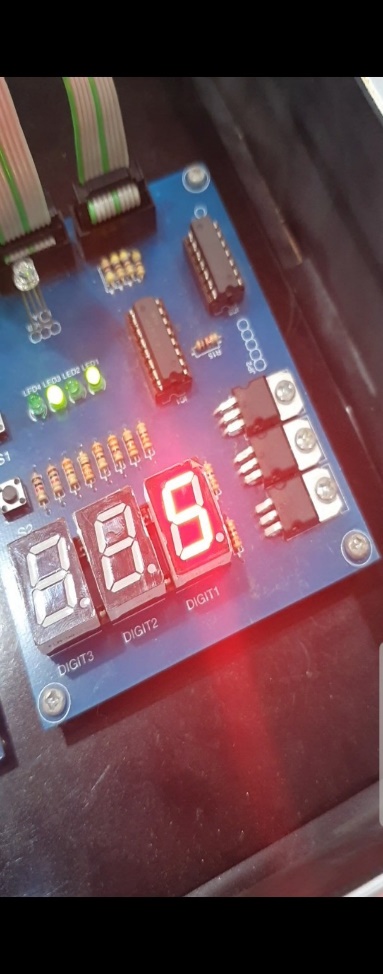
Dalam praktikum ini, arduino digunakan untuk membaca input dari keypad lalu menampilkannya pada 3 buah seven segment. Namun, karena jumlah pin arduino yang terbatas digunakan decoder 74LS47 untuk mengurangi pin seven-segment (7 pin) menjadi 4 pin, decoder 7402N untuk multiplexing pada ketiga seven-segment, dan encoder 74C922 untuk mengurangi pin keypad (16 pin) menjadi 4 pin.

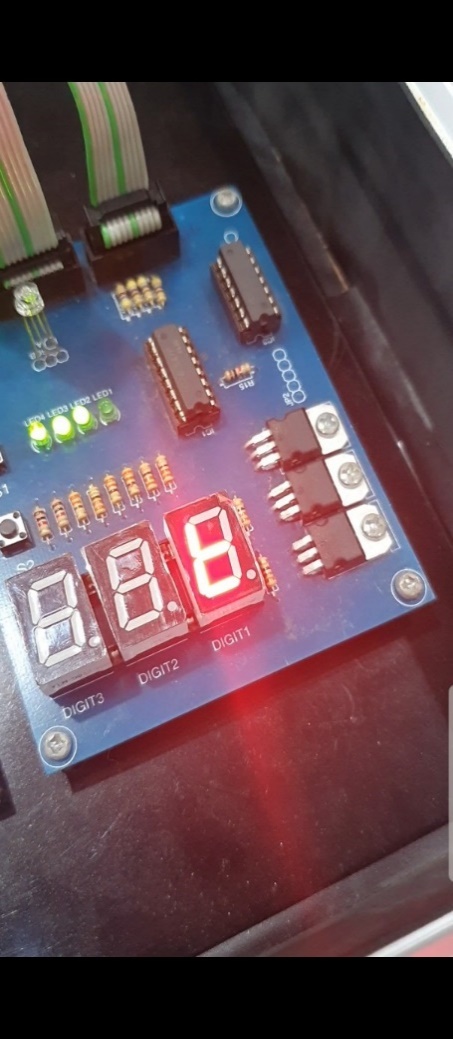
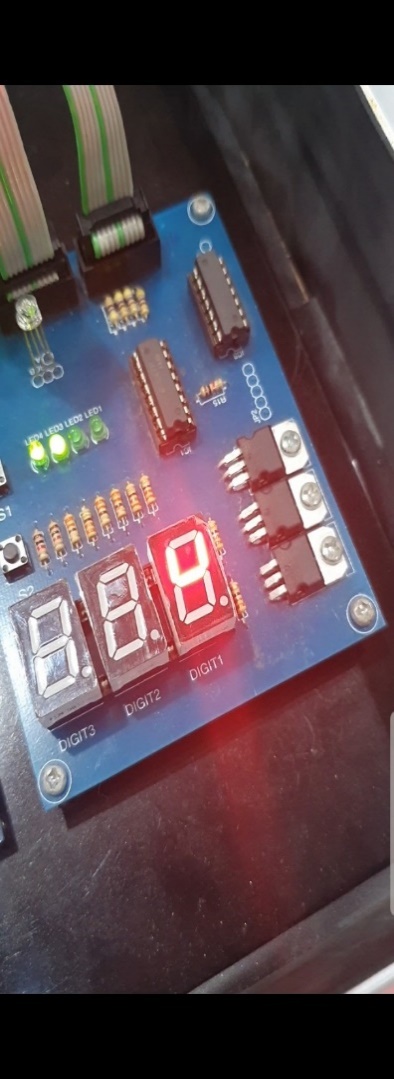
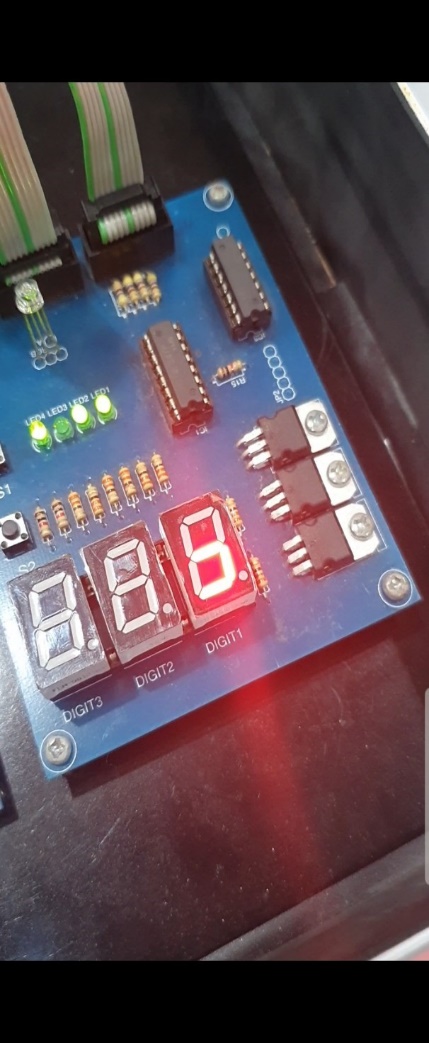
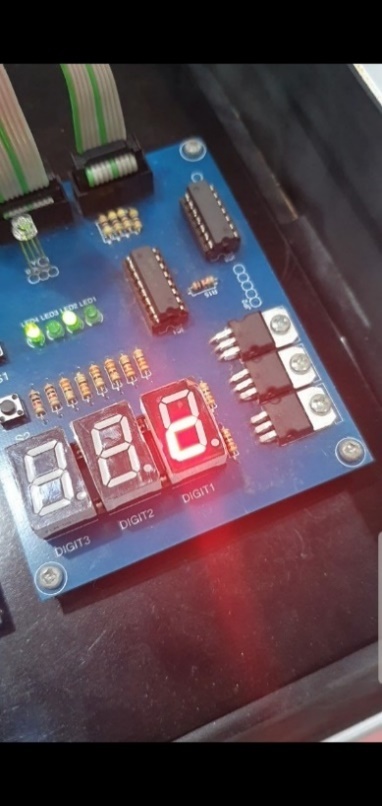
# Data dan Analisis

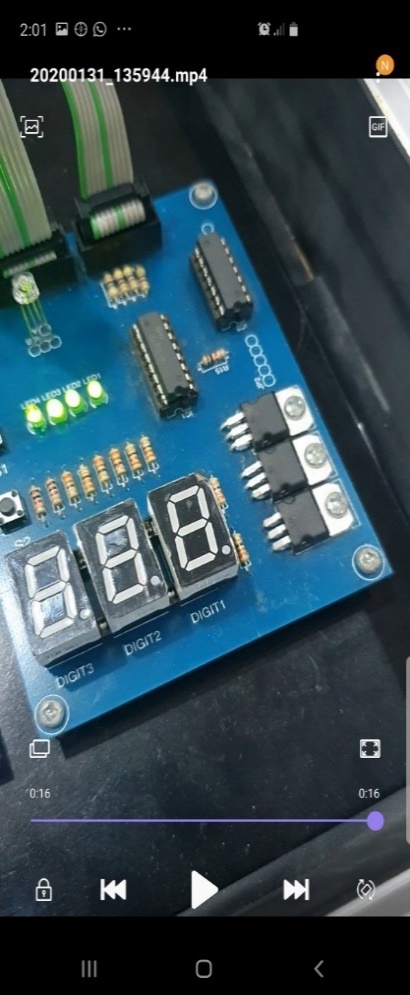
## Seven Segment Counter

Pada percobaan ini, arduino menggunakan program Modul1\_test.ino (terlampir) yang menampilkan angka pada seven-segment. Dapat dilihat bahwa tampilan pada seven-segment ketika diberi input sesuai dengan datasheet 74LS47.

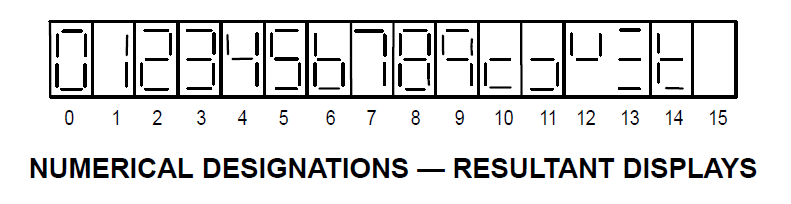








Gambar 1 Tampilan seven-segment dari input 0 sampai 15 (berurutan)



Gambar 2 Tampilan seven-segment menurut Datasheet

Sumber: http://www.applelogic.org/files/74LS47.pdf

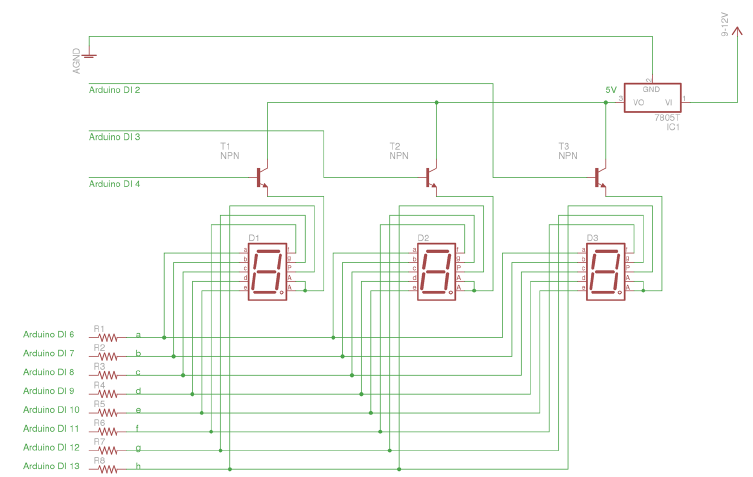
Tampilan seven-segment juga menghasilkan nyala yang terang dengan tingkat kecerahan sama pada setiap tampilan angka. Penulis juga membandingkan data ini dengan kelompok lain dan mendapatkan semua data pembanding menghasilkan tingkat kecerahan yang bagus dan stabil pada tiap angka.

Tabel 1 Tingkat kecerahan LED seven-segment

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kelompok | Tampilan angka pada seven-segment | | | | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 18\* | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 10 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 11 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 13 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 19 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |

\* kelompok penulis

Kecerahan yang stabil pada tiap tampilan angka ini disebabkan oleh konfigurasi rangkaian antara decoder 74LS47 dengan seven-segment. Resistor pembatas arus dipasangkan pada tiap katoda, bukan hanya memasang 1 pada anoda. Hal ini menyebabkan arus yang melewati tiap LED tetap konstan, yang bertambah hanya arus yang dialirkan arduino. Sementara, jika hanya memasang 1 resistor pada anoda, arus yang dialirkan arduino akan konstan. Sehingga, semakin banyak LED yang menyala, semakin kecil pula arus yang mengalir pada tiap LED. Ini akan membuat tampilan angka dengan banyak LED (misalnya 8) lebih redup dari angka dengan sedikit LED (misalnya 1).



Gambar 3 Skematik rangkaian pada seven-segment

Sumber: Mursito, E., & Mahyuddin, H. (2019). Modul 1 - Digital Input Output. Bandung: TF ITB.

## State Driven dan Event Driven Counter

Pada praktikum ini, dibuat program pencacah (counter). Pencacahan dilakukan dengan 2 tombol input, tombol pertama untuk melaksanakan pencacahan, tombol kedua untuk mengubah mode pencacahan naik atau turun. Dapat dilihat program berjalan dengan baik dan bekerja sesuai harapan.

Tabel 2 Output program pencacah

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Perlakuan | State-driven | | Event-driven | |
| Yang Diharapkan | Hasil | Yang Diharapkan | Hasil |
| 1 | Arduino mulai | LED1=0 | 0 | LED1=0 | 0 |
| 2 | Tekan S1 | LED1=1 | 1 | LED1=1 | 1 |
| 3 | Tekan S1 8 kali | LED1=9 | 9 | LED1=9 | 9 |
| 4 | Tekan S1 sekali lagi | LED1=0 | 0 | LED1=0 | 0 |
| 5 | Tekan S2 | dalam kondisi (state down) | tidak ada perubahan (state down) | kondisi state down | tidak ada perubahan (state down) |
| 6 | Tekan S1 | LED1=9 | 9 | LED1=9 | 9 |
| 7 | Tekan S1 8 kali | LED! = 1 | 1 | LED1=1 | 1 |
| 8 | Tekan S1 sekali lagi | LED1 = 0 | 0 | LED1=0 | 0 |
| 9 | Tekan S2 | dalam kondisi state up | state up | state up | state up |
| 10 | Tekan S1 | LED1= 1 | 1 | LED1=1 | 1 |

Program untuk melaksanakan ini dapat dilakukan dengan 2 metode, yaitu state-driven dan event-driven. Perbedaan kedua metode ini terletak pada level dimana input masuk. Pada state-driven, program sudah berada pada salah satu state bahkan sebelum diberi input. Sementara, pada event-driven program baru melaksanakan sebuah event ketika diberikan input.

A close up of a clock

Description automatically generated

Gambar 4 Flowchart program pencacah state-driven

A close up of a map

Description automatically generated

Gambar 5 Flowchart program pencacah event-driven

## Shifting Display dengan Multiplexing

Menampilkan 3 digit angka pada ketiga seven-segment tidak bisa dilakukan secara bersamaan. Hal ini karena pin arduino hanya terhubung pada 1 decoder 74LS47 yang terhubung pada ketiga seven-segment sekaligus. Sehingga, teknik yang digunakan adalah multiplexing dengan frekuensi tinggi. Yaitu, tampilan pada seven-segment muncul bergantian pada tiap layar seven-segment namun dilakukan dengan sangat cepat sehingga terlihat seakan-akan ketiga seven-segment menampilkan angka secara bersamaan.

Tabel 3 Display periode optimum

|  |  |  |
| --- | --- | --- |
| Kelompok | Display periode (ms) | Frekuensi (Hz) |
| 18\* | 7 | 142,86 |
| 10 | 8 | 125 |
| 11 | 5 | 200 |
| 13 | 1 | 1000 |
| 19 | 8 | 125 |

\* kelompok penulis

Dapat dilihat bahwa rata-rata kelompok praktikan menggunakan display periode sebesar 5,8 ms atau sekitar 172,41 Hz. Perbedaan nilai ini akibat persepsi mata manusia yang beragam pada tiap individu. Secara umum, manusia dapat mendeteksi secara fisiologis hingga 1000 frame per second (fps) namun hanya dapat mendeteksi secara sadar hingga 150 fps[[3]](#footnote-3). Perhatikan bahwa rata-rata display periode yang digunakan yaitu 172,41 fps lebih tinggi dari nilai referensi ini.

## Stopwatch

Pada bagian praktikum ini, dibuat program untuk menghitung waktu (stopwatch) dengan 2 tombol input. Tombol pertama adalah start/stop, dan tombol kedua adalah reset. Program dibuat seperti pada Gambar 6 Flowchart program stopwatch.

A close up of a map

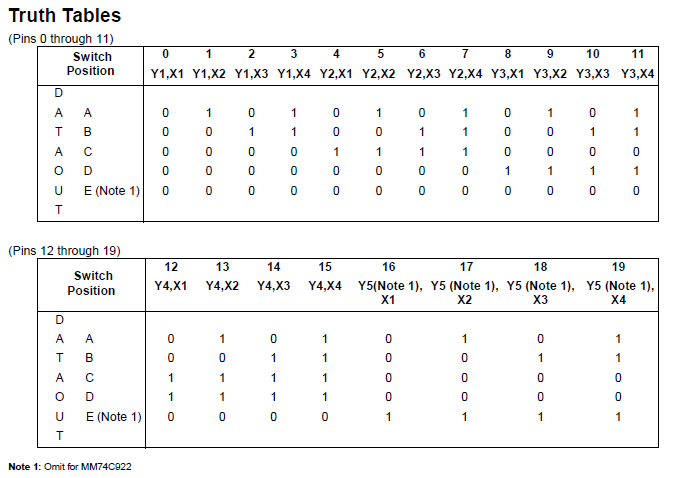
Description automatically generated

Gambar 6 Flowchart program stopwatch

Perhatikan bahwa digunakan fungsi millis() untuk membaca internal clock arduino. Hal ini agar program dapat mengatur prioritas untuk fungsi yang dijalankan, sehingga program tidak terlambat dalam counting stopwatch yang harus tepat dalam satuan detik. Didapatkan program stopwatch berjalan dengan baik dan menghitung waktu dengan cukup tepat ketika dibandingkan dengan stopwatch smartphone.

## Keypad

Untuk membuat arduino dapat menerima input angka, digunakan keypad 4x4 dengan encoder 74C922. Karena sinyal yang dibaca arduino dari encoder berupa biner, maka dibuat suatu “kamus” dalam bentuk array pada program untuk menerjemahkan biner tersebut ke integer. Namun, setelah dilakukan penyesuaian dan kalibrasi, didapatkan bahwa sinyal yang diterima arduino tidak cocok dengan datasheet 74C922. Hal ini dapat diakibatkan kesalahan komponen yang digunakan, yaitu komponen encoder bukanlah 74C922, namun hal ini sepertinya tidak mungkin. Sehingga, penulis menyimpulkan bahwa terjadi kesalahan pemasangan pin pada PCB kit praktikum sehingga pembacaan keypad berbeda dengan datasheet.



Gambar 7 Tabel kebenaran encoder 74C922 menurut datasheet

Sumber: <https://www.soemtron.org/downloads/disposals/74c922.pdf>

Tabel 4 Penerjemahan biner output decoder 74C922

|  |  |
| --- | --- |
| Menurut datasheet | Setelah kalibrasi |
| 0 | 0xD |
| 1 | 0xE |
| 2 | 0 |
| 3 | 0xF |
| 4 | 0xC |
| 5 | 9 |
| 6 | 8 |
| 7 | 7 |
| 8 | 0xB |
| 9 | 6 |
| 10 | 5 |
| 11 | 4 |
| 12 | 0xA |
| 13 | 3 |
| 14 | 2 |
| 15 | 1 |

## Calculator

Pada bagian praktikum ini dibuat program yang dapat menerima input angka dari keypad dan dapat mengoperasikannya sebagaimananya kalkulator biasa. Namun karena tampilan yang terbatas pada 3 digit (3 buah seven-segment), kalkulator mempunyai batas nilai tertinggi 999. Program dibuat seperti pada Gambar 8 Flowchart program kalkulator.

A close up of a map

Description automatically generated

Gambar 8 Flowchart program kalkulator

Karena keypad berisikan tombol angka 0 - 9, A – D, \*, #, maka ditetapkan secara beruturut-turut A, B, C, dan D sebagai operator +, -, ×, dan ÷. Sementara tombol \* ditetapkan sebagai clear dan # sebagai calc. Didapatkan program bekerja sesuai harapan. Namun akibat keypad yang cukup sensitif dan seringkali stuck, operasi calc sering gagal karena input yang overloaded (misalnya menekan angka 5 sekali namun terbaca 555).

# Kesimpulan

1. Didapatkan tingkat kecerahan LED seperti pada Tabel 1 Tingkat kecerahan LED seven-segment
2. Ditentukan program pencacah seperti pada Gambar 4 Flowchart program pencacah state-driven dan Gambar 5 Flowchart program pencacah event-driven. Kode program terlampir (Modul1\_1.ino, Modul1\_2.ino)
3. Didapatkan nilai display periode yang optimal sebesar 7 ms. Adapun data pembanding tertera pada Tabel 3 Display periode optimum
4. Ditentukan program stopwatch seperti pada Gambar 6 Flowchart program stopwatch. Kode program terlampir (Modul1\_4.ino)
5. Ditentukan tabel translasi yang benar seperti pada Tabel 4 Penerjemahan biner output decoder 74C922
6. Ditentukan program kalkulator seperti pada Gambar 8 Flowchart program kalkulator. Kode program terlampir (Modul1\_5.ino)

# Daftar Pustaka

*74C922 Datasheet.* (2020, Februari 1). Diambil kembali dari The Soemtron Calculator Collection: https://www.soemtron.org/downloads/disposals/74c922.pdf

*74LS47 Datasheet.* (2020, Februari 1). Diambil kembali dari AppleLogic: http://www.applelogic.org/files/74LS47.pdf

*Arduino Uno*. (2020, Februari 2). Diambil kembali dari Components101 - Electronic Components Pinouts, Details & Datasheets: https://components101.com/microcontrollers/arduino-uno

Mursito, E., & Mahyuddin, H. (2019). *Modul 1 - Digital Input Output.* Bandung: TF ITB.

Rafiquzzaman, M. (2014). *Fundamentals of Digital Logic and Microcontrollers : Sixth Edition.* New Jersey: Wiley.

*What is the highest frame rate (fps) that can be recognized by human perception. At what rate do we essentially stop noticing the difference*. (2020, Februari 4). Diambil kembali dari Quora: https://www.quora.com/What-is-the-highest-frame-rate-fps-that-can-be-recognized-by-human-perception-At-what-rate-do-we-essentially-stop-noticing-the-difference

# Lampiran

Daftar File Terlampir:

1. Modul1\_test.ino
2. Modul1\_1.ino
3. Modul1\_2.ino
4. Modul1\_3.ino
5. Modul1\_4.ino
6. Modul1\_keypad.ino
7. Modul1\_5.ino

1. Arduino Uno, <https://components101.com/microcontrollers/arduino-uno> [↑](#footnote-ref-1)
2. Rafiquzzaman, M. (2014). *Fundamentals of Digital Logic and Microcontrollers ; Sixth Edition*. New Jersey: Wiley. [↑](#footnote-ref-2)
3. Quora: https://www.quora.com/What-is-the-highest-frame-rate-fps-that-can-be-recognized-by-human-perception-At-what-rate-do-we-essentially-stop-noticing-the-difference [↑](#footnote-ref-3)