



LAPORAN HASIL PENELITIAN

KOMPARASI HASIL CACAH PULSA ANTARA METODE *INTERRUPT* DAN *TIMER/COUNTER* PADA STM32F103C8T6

Oleh :

**Ir. Mohammad Amin HD, MT
Akality Mardhiya, ST, MT**

PROGRAM STUDI SISTEM KOMPUTER

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS BOROBUDUR
JAKARTA**

2021



LAPORAN HASIL PENELITIAN

KOMPARASI HASIL CACAH PULSA ANTARA METODE *INTERRUPT* DAN *TIMER/COUNTER* PADA STM32F103C8T6

Oleh :

Ir. Mohammad Amin HD, MT

Akalily Mardhiya, ST, MT

PROGRAM STUDI SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BOROBUDUR

2021

**LEMBAR IDENTITAS DAN PENGESAHAN
LAPORAN AKHIR PENELITIAN**

1	Judul Penelitian	Komparasi Hasil Cacah Pulsa antara Metode <i>Interrupt</i> dan <i>Timer/Counter</i> pada STM32f103c8t6
2	Ketua Peneliti :	
	a. Nama	Ir. Mohammad Amin HD, MT
	b. Jenis Kelamin	Laki-Laki
	c. Pangkat/Golongan/NIP	-
	d. Jabatan Fungsional	-
	e. Fakultas/Program Studi	Fakultas Ilmu Komputer/Sistem Komputer
	f. Bidang ilmu yang diteliti	Sistem Komputer
3	Jumlah Tim Peneliti	2 (dua) orang
4	Lokasi Penelitian	Jakarta
5	Jangka Waktu Penelitian	6 (enam) bulan
6	Biaya diperlukan	Rp. 10.000.000,-
7	Sumber Dana	Mandiri

Jakarta, 21 Januari 2021

Mengetahui
Fakultas Ilmu Komputer
Universitas Borobudur
Dekan



Prof. Dr. H. Suryanto, MM

Ketua Peneliti

Ir. Mohammad Amin HD, MT

Kepala Lembaga Penelitian dan Pengabdian Masyarakat
LPPM – Universitas Borobudur



Evi Syafriada Nasution, S.Psi, M.Psi

RINGKASAN

Pencacah pulsa adalah sebuah modul elektronika yang dipergunakan untuk menghitung jumlah kejadian persatuan waktu. Modul tersebut banyak dijumpai di dalam kehidupan sehari-hari, seperti penghitung besar kecepatan kendaraan, besar tekanan darah dan denyut nadi, besar kecepatan angin, nilai suhu, tekanan, dan kelembaban, dan lain-lain.

Perkembangan teknologi digital dan pemroses mikro yang semakin kecil dan murah; seperti mikrokontroler dan FPGA; membuka peluang dalam menyederhanakan pembuatan modul-modul elektronika, tak terkecuali modul pencacah pulsa. Modul pencacah pulsa yang sebelumnya menggunakan sejumlah komponen digital yang tersusun dalam bentuk untaian rangkaian flip-flop, saat ini dapat menggunakan sebuah sistem minimum dari mikrokontroler.

Hingga saat ini telah banyak dijumpai sistem minimum dari jenis mikrokontroler berbeda. Jenis-jenis mikrokontroler tersebut antara lain, Intel MCS-51, AVR, ARM, dan PIC. Salah satu sistem minimum mikrokontroler yang menarik untuk diteliti adalah yang menggunakan mikrokontroler dari jenis ARM. Dari sistem minimum berbasis mikrokontroler ARM ini, kami memilih sistem minimum yang banyak tersedia di pasaran, harga murah, memiliki kapasitas memori yang cukup tinggi, clock prosesor yang lebih cepat, dan tentunya sistem tersebut menyediakan fitur interupsi dan timer/counter. Sistem minimum itu adalah STM32F103C8T6.

Kegiatan penelitian adalah melakukan komparasi hasil cacah pulsa dari STM32F103C8T6 dengan metode interupsi dan timer/counter. Tujuan dari penelitian ini adalah untuk mengetahui metode cacah pulsa yang terbaik dalam merespon cacah pulsa yang memiliki kejadian/event pada level jangkauan perdetik detik hingga per 10 mikrodetik.

Pengujian terhadap respon kejadian dilakukan dengan menggunakan alat pembangkit cacah pulsa yang memiliki kecepatan pulsa yang dapat diatur. Hasil percobaan menunjukkan bahwa baik metode interupsi maupun metode timer/counter dapat melakukan pencacahan pulsa tanpa menunjukkan perbedaan diantara kedua metode tersebut secara nyata.

Kata Kunci : STM32F103C8T6, interupsi, timer/counter.

PRAKATA

Dengan memanjatkan puji syukur ke hadirat Allah SWT, karena dengan rahmat, hidayah, dan karuniaNya telah memperkenankan penulis untuk menyelesaikan Penelitian yang berjudul “KOMPARASI HASIL CACAH PULSA ANTARA METODE *INTERRUPT* DAN *TIMER/COUNTER* PADA STM32F103C8T6”. Selama melaksanakan penelitian ini, banyak sekali bantuan dan dukungan yang telah diperoleh. Untuk itu pada kesempatan ini, penulis bermaksud mengucapkan terima kasih kepada beberapa pihak yang diantaranya adalah :

1. Bapak Prof. Dr. H. Suryanto, MM, selaku Dekan Fakultas Ilmu Komputer Universitas Borobudur Jakarta
2. Ibu Evi Syafrida Nasution, S.Psi, M.Psi selaku Ketua LPPM Universitas Borobudur Jakarta.
3. Rekan-rekan dosen Fakultas Ilmu Komputer Universitas Borobudur yang telah berbagi ilmu dalam menjalankan penelitian.

Sungguh penelitian ini bukanlah tanpa kelemahan, untuk itu kritik dan saran yang bersifat konstruktif sangat diharapkan. Akhir kata, semoga laporan penelitian ini dapat bermanfaat.

Jakarta, Januari 2021

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
RINGKASAN	iii
PRAKATA	iv
DAFTAR ISI.....	v
BAB I PENDAHULUAN.....	1
BAB II TINJAUAN PUSTAKA.....	2
2.1. Microcontroler STM32F103C8T6.....	2
2.1.1. Tinjauan MCU	2
2.1.2. Kelompok Keluarga STM32F103.....	3
2.2. Interrupsi pada Mikrokontroler.....	4
2.3. Timer/Counter	5
2.4 Komunikasi I2C	5
2.5. Komunikasi UART	7
2.6.Liquid Crystal Display (LCD) 16 x 4 dengan I2C.....	7
BAB III TUJUAN DAN MANFAAT PENELITIAN	8
3.1 Maksud dan Tujuan Penelitian	8
3.2 Manfaat Penelitian.....	8
BAB IV METODE PENELITIAN	9
4.1 Metode Penelitian.....	9
4.2. Alat dan Bahan.....	11
4.3. Langkah-langkah Penelitian.....	15
BAB V HASIL DAN PEMBAHASAN.....	16
5.1. Hasil Konfigurasi Mikrokontroler.....	16
5.2. Hasil Implementasi Program	16
5.3. Analisis Hasil Komparasi.....	17
BAB VI KESIMPULAN DAN SARAN	19
6.1. Kesimpulan.....	19
6.2. Saran.....	19

DAFTAR PUSTAKA	20
LAMPIRAN 1	21
LAMPIRAN 2	22
LAMPIRAN 3	33
LAMPIRAN 4	46
L4.1 Potongan Program Utama Mode Interrupt	46
L4.2 Potongan Program Utama Mode Timer/Counter.....	46
L4.3. Potongan Program Mode Cacah Pulsa Interrupt	47
L4.4. Potongan Program Mode Cacah Pulsa Timer/Counter.....	47
LAMPIRAN 5	48

BAB I

PENDAHULUAN

Pencacah pulsa adalah sebuah modul elektronika yang dipergunakan untuk menghitung jumlah kejadian persatuan waktu. Modul tersebut banyak dijumpai di dalam kehidupan sehari-hari, seperti penghitung besar kecepatan kendaraan, besar tekanan darah dan denyut nadi, besar kecepatan angin, nilai suhu, tekanan, dan kelembaban, dan lain-lain.

Perkembangan teknologi digital dan pemroses mikro yang semakin kecil dan murah; seperti mikrokontroler dan FPGA; membuka peluang dalam menyederhanakan pembuatan modul-modul elektronika, tak terkecuali modul pencacah pulsa. Modul pencacah pulsa yang sebelumnya menggunakan sejumlah komponen digital yang tersusun dalam bentuk untaian rangkaian flip-flop, saat ini dapat menggunakan sebuah sistem minimum dari mikrokontroler.

Hingga saat ini telah banyak dijumpai sistem minimum dari jenis mikrokontroler berbeda. Jenis-jenis mikrokontroler tersebut antara lain, Intel MCS31, MCS51, AVR, ARM, dan PIC. Salah satu sistem minimum mikrokontroler yang menarik untuk diteliti adalah yang menggunakan mikrokontroler dari jenis ARM. Dari sistem minimum berbasis mikrokontroler ARM ini, kami memilih sistem minimum yang banyak tersedia di pasaran, harga murah, memiliki kapasitas memori yang cukup tinggi, clock prosesor yang lebih cepat, dan tentunya sistem tersebut menyediakan fitur interupsi dan timer/counter. Sistem minimum itu adalah STM32F103C8T6.

Tujuan dari kegiatan penelitian ini adalah membuat pencacah pulsa berbasis STM32F103C8T6 dengan metode interupsi dan timer/counter. Tujuan penelitian tersebut adalah untuk mendapatkan jawaban tentang bagaimana membuat sebuah pencacah pulsa dengan metode interupsi dan timer/counter pada mikrokontroler STM32F103C8T6. Sementara itu, maksud dari penelitian ini adalah untuk mengetahui ada atau tidaknya perbedaan secara nyata hasil cacah pulsa antara metode interupsi dan metode timer/counter.

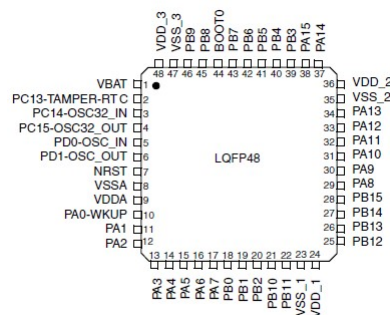
BAB II TINJAUAN PUSTAKA

2.1. Microcontroller STM32F103C8T6

STM32 adalah keluarga rangkaian terpadu dari mikrokontroler 32-bit keluaran STMicroelectronics. Chip STM32 dikelompokkan ke dalam seri terkait yang berbasis di sekitar inti prosesor ARM 32-bit yang sama, seperti Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+, atau CortexM0. Secara internal, setiap mikrokontroler terdiri dari minimal sebuah CPU, memori RAM statis, memori flash, antarmuka debugging, dan berbagai periferal pembantu [1].

2.1.1 Tinjauan MCU

Keluarga Mikrokontroler STM32F103xx terdiri dari ARM Cortex-M3 32-bit *RISC* inti, memori tertanam berkecepatan tinggi (memori Flash hingga 128 Kbytes dan *Static RandomAccess Memory* (SRAM) hingga 20 Kbytes), I/O (Input/ Output) dan periferal yang merekabekerja sama dengan menghubungkan ke dua bus *Advanced Peripheral Bus* (APB). Mikrokontroler STM32F103xx mencakup banyak periferal serta dua ADC 12 bit, sebuah *Advanced Control Timer*, tiga timer 16-bit General Purpose dan juga *Pulse WidthModulasi* (PWM) pengatur waktu. Ini juga disediakan oleh dua I²Cs (Inter-Integrated Circuit) dan SPI (Serial Peripheral Interface), tiga *Universal Synchronous/Asynchronous ReceiveTransmitter* (USART), USB dan CAN (Controller Area Network) sebagai antarmuka komunikasisistem. Gambar 2.1 menyajikan pinout untuk keluarga STM32F103 yang digunakan dalam penelitian ini.



Gambar 2.1 Pinout STM32F103

Mikrokontroler berdensitas menengah digunakan dan memiliki 64 pin. Keluarga mikrokontroler ini memiliki tiga port yakni, PA, PB dan PC. Masing-masing dari port-port MCU tersebut memiliki 16 pin untuk jalur I/O. Sementara itu, pin Vss, Vdd dan Vbatt diperlukan untuk melakukan bias tegangan ke mikrokontroler dengan menggunakan catu daya eksternal.

2.1.2. Kelompok Keluarga STM32F103

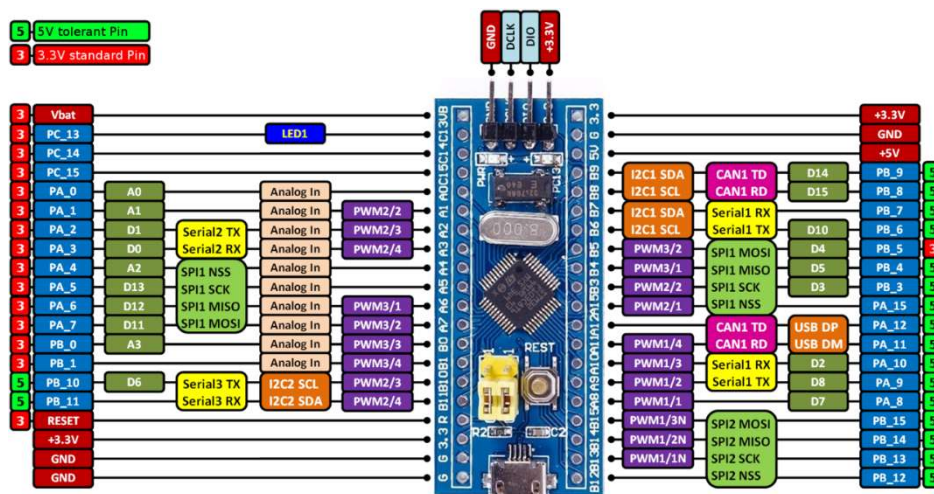
Mikrokontroler keluarga STM32F103xx terdiri dari tiga kelompok:

1. Kerapatan rendah: STM32F103x4 dan STM32F103x6 adalah perangkat berdensitas rendah.
2. Kerapatan sedang: STM32F103x8 dan STM32F103xB adalah Perangkat dengan kerapatan sedang.
3. Kerapatan tinggi: STM32F103xC, STM32F103xD dan STM32F103xE adalah perangkat dengan kerapatan tinggi.

Ketiga kelompok ini dibuat sesuai dengan fitur mikrokontroler STM32F103xx anggota keluarga. Mikrokontroler berdensitas rendah memiliki memori Flash dan RAM yang lebih rendah (Akses Acak Memori), timer kurang dan periferal di bandingkan dengan dua kelompok lainnya. Dengan demikian, Menengah- kepadatan dan High-density terdiri dari memori Flash yang lebih tinggi, kapasitas RAM dan juga lebih banyak periferal tambahan. Keluarga dengan kepadatan rendah mencakup 16 KB hingga 32 KB memori Flash dan 6 KB hingga 10 KB RAM kapasitas. Mereka terdiri dari 1 × BISA, 1 × USB, 1 × PWM timer, 1 × I²C, 1 × SPI dan 2 × ADC, 2 × USART, dan 2 × 16-bit timer. Perbedaan antara keluarga berkepadatan rendah berkenaan dengan jumlah paket pinout mereka. Ada tiga jenis paket pinout yang mereka buat 36, 48 dan 64 pin. Memori Flash mereka ditingkatkan dari 64 KB menjadi 128KB kapasitas KB dan RAM adalah 20 KB. Keluarga dengan kepadatan menengah memiliki lebih banyak properti dibandingkan dengan keluarga dengan kepadatan rendah. Itujumlah periferal dan pinouts dari mereka ditingkatkan. Mereka memiliki 1 × BISA, 1 × USB, 1 × timer PWM, 2 × I²C, 2 × SPI, 2 × ADC, 3 × USART, dan 3 × 16-bit timer. Mereka jugamemiliki tiga jenis paket pinout yang terdiri dari 48, 64 dan 100 pin. Keluarga dengan kepadatan tinggi diselesaikan lebih dari yang lain

dan memiliki lebih banyak periferal. Mereka membuatup periferal ini, seperti 1 × BISA, 1 × USB, 1 × PWM timer, 2 × I²Ss (SPI), 2 × I²C, 3 × SPI, 2 × ADC, 1 × DAC, 5 × USART, 2 × timer dasar, 4 × 16-bit timer, 1 × SDIO(Output Input Digital Aman), dan 1 × FSMC (Pengontrol Memori Statis Fleksibel).

Gambar 2.2. memperlihatkan konfigurasi kaki-kaki salah satu keluarga STM32F103 dengan kerapatan sedang, yaitu STM32F103C8T6.



Gambar 2.2. Konfigurasi kaki STM32F103C8T6

2.2. Interrupsi pada Mikrokontroler

Interupsi adalah proses dalam sistem mikrokontroler yang menghentikan aliran program utama akibat terjadinya (event) trigger (pemicu) tertentu dari suatu sumber (vector) interupsi dan memaksa sistem mikrokontroler untuk mengeksekusi sub-rutin/fungsi/blok program layanan interupsi (interrupt service routine, ISR) hingga selesai (complete). Setelah program interupsi selesai dikerjakan, maka sistem mikrokontroler akan kembali melanjutkan program utama yang sebelumnya dihentikan. Pada sebuah komponen embedded systems seperti mikrokontroler, memiliki 2 jenis sumber interupsi, yaitu:

1. Interrupsi internal Sumber (vektor) interupsi yang berasal dari dalam sistem mikrokontroler itu sendiri.
2. Interrupsi eksternal Sumber interupsi yang berasal dari luar sistem mikrokontroler.

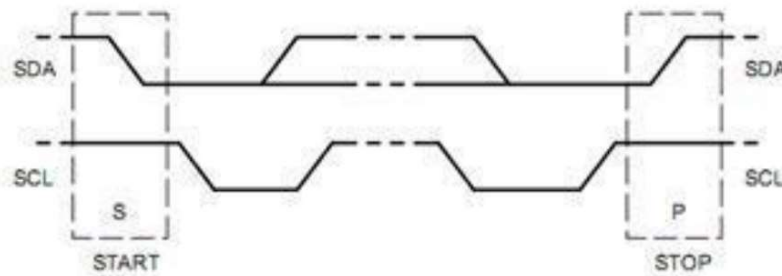
2.3. Timer/Counter

Timer/Counter adalah suatu periferal yang tertanam didalam microcontroller yang berfungsi sebagai pewaktu. Dengan periferal ini pengguna microcontroller dapat dengan mudah menentukan kapan suatu perintah dijalankan (delay), tentu saja fungsi timer tidak hanya untuk penundaan perintah saja, timer juga dapat berfungsi sebagai oscilator, PWM, ADC, dan lain-lain. Cara kerja timer adalah dengan cara memberikan prescaling (membagi frekuensi) pada clock yang terpasang pada microcontroller sehingga timer dapat berjalan dengan frekuensi yang diinginkan. Tujuan Penggunaan Timer & Counter:

1. Melaksanaan tugas secara ber-ulang
2. Mengendalikan kecepatan motor DC (PWM).
3. Membuat pencacahan (Counter)
4. Membuat penundaan waktu (delay)

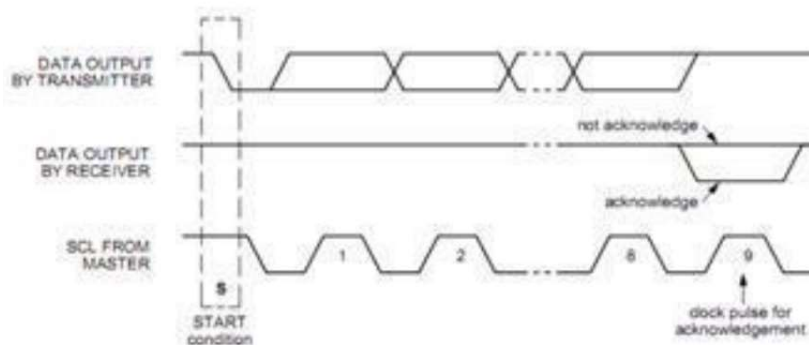
2.4. Komunikasi I2C

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari kanal SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dioperasikan sebagai pasangan Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan memunculkan sinyal clock. Slave adalah piranti yang dialamati master. Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "1" menjadi "0" pada saat SCL "1". Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "0" menjadi "1" pada saat SCL "1". Kondisi sinyal Start dan sinyal Stop seperti tampak pada Gambar 2.3.



Gambar 2.3. Kondisi sinyal start dan stop.

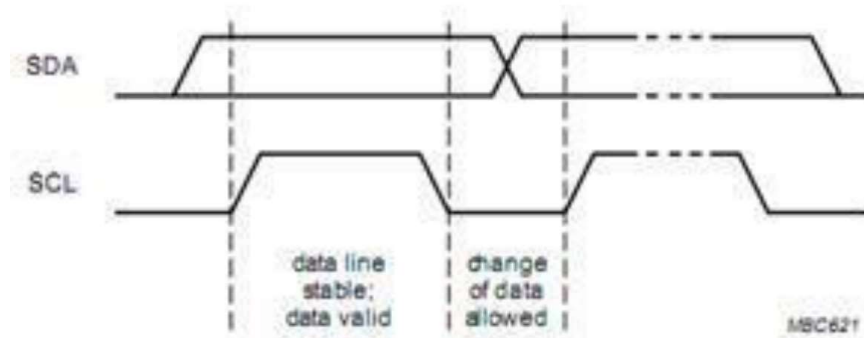
Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK. Setelah transfer data oleh master berhasil diterima slave, slave akan menjawab dengan mengirim sinyal *acknowledge*, yaitu dengan menyet jalur SDA ke “0” selama siklus clock ke 9. Ini mengindikasikan bahwa slave telah menerima 8 bit data dari master. Kondisi sinyal *acknowledge* seperti tampak pada Gambar 2.4.



Gambar 2.4. Sinyal ACK dan NACK.

Dalam melakukan transfer data pada I2C Bus (Gambar 2.5), kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- ✓ Transfer data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- ✓ Selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal Start atau sinyal Stop .



Gambar 2.5. Transfer Bit pada I2C bus

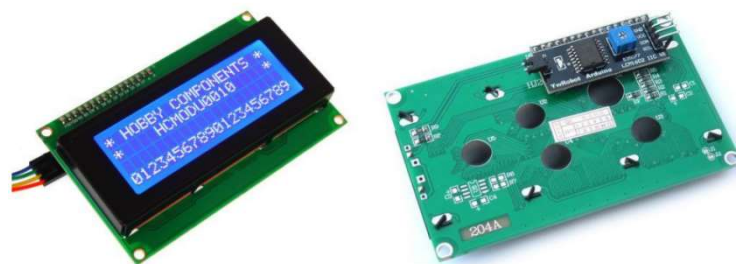
2.5. Komunikasi UART

UART atau Universal Asynchronous Receiver-Transmitter adalah bagian perangkat keras komputer yang menerjemahkan antara bit-bit paralel data dan bit-bit serial. UART biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi serial pada komputer atau port serial perangkat peripheral.

2.6. Liquid Crystal Display (LCD) 16 x 4 dengan I2C

LCD (Liquid Crystal Display) adalah perangkat yang berfungsi sebagai media penampil dengan memanfaatkan kristal cair sebagai objek penampil utama. LCD sering dijumpai pada beberapa peralatan, seperti media elektronik televisi, kalkulator, atau layar komputer.

LCD yang digunakan dalam penelitian ini adalah LCD berukuran 20x4 karakter (Gambar 2.6) dengan tambahan driver I2C yang berfungsi sebagai modul konverter sinyal komunikasi dari paralel ke serial. Penggunaan driver I2C tersebut adalah untuk menghemat penggunaan jalur komunikasi menjadi hanya 4 jalur, yaitu jalur SCL, SDA, Vcc dan GND.



Gambar 2.6. LCD 20x4 dengan driver I2C

BAB III

TUJUAN DAN MANFAAT PENELITIAN

3.1 Maksud dan Tujuan Penelitian

Maksud adalah untuk mendapatkan informasi ada tidaknya perbedaan hasil cacah pulsa dengan melakukan komparasi hasil cacah pulsa yang diperoleh dari mikrokontroler STM32F103C8T6 antara hasil cacah yang menggunakan metode *Interrupt* dan hasil cacah yang menggunakan metode *Counter/Timer*.

Tujuan penelitian adalah membuat modul pencacah pulsa berbasis STM32F103C8T6 menggunakan metode Interupsi dan metode Timer/Counter.

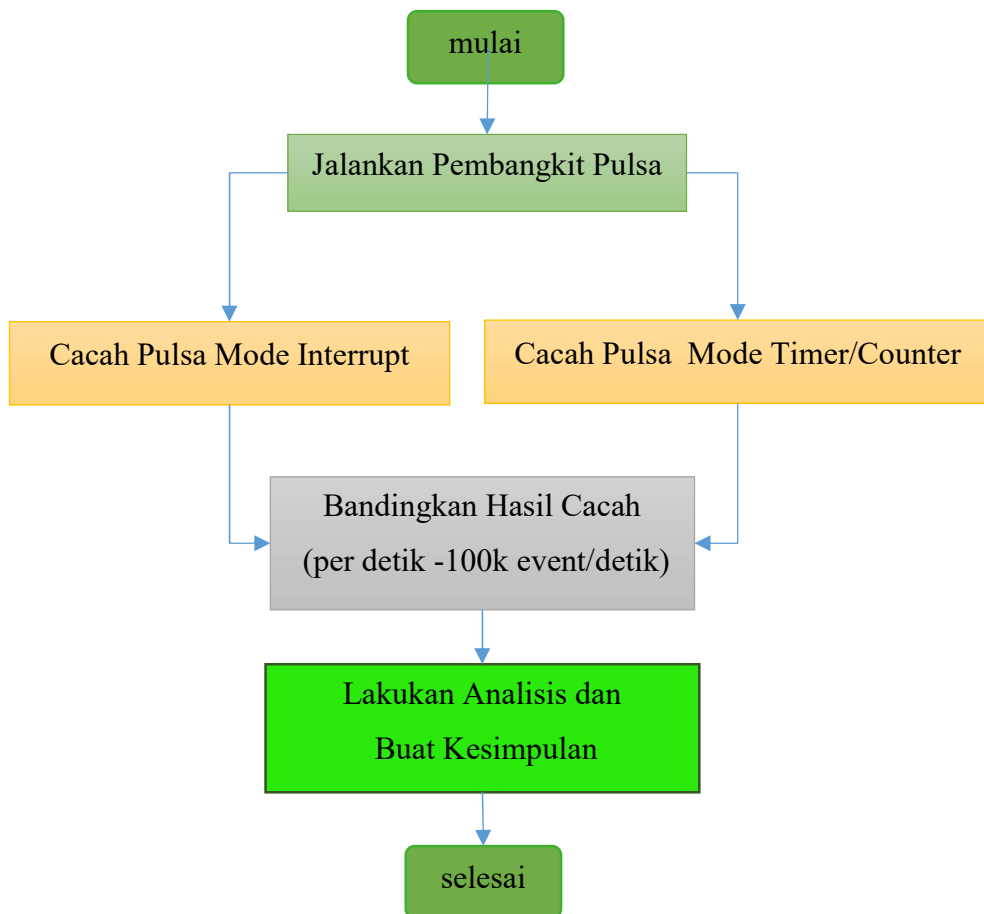
3.2 Manfaat Penelitian

Manfaat dari hasil penelitian ini adalah untuk mendapatkan informasi tetnang metode pencacahan pulsa pada STM32F103C8T6 yang paling sesuai digunakan pada pencacahan pulsa atau kejadian dalam rentang dari kejadian per detik hingga kejadian per mikrodetik.

BAB IV METODE PENELITIAN

4.1 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode komparasi antara hasil cacah pulsa menggunakan teknik interupsi dan hasil cacah pulsa menggunakan teknik counter/timer. Secara sederhana, metode penelitian ini diperlihatkan dalam Gambar 4.1.



Gambar 4.1. Metode Komparasi Hasil Cacah

Metode komparasi yaitu suatu cara analisis untuk mendapatkan informasi tentang ada atau tidaknya perbedaan dari dua atau lebih kelompok data yang sedang dibandingkan. Dalam penelitian ini, kelompok data yang akan di

bandingkan adalah data hasil cacah pulsa yang diperoleh dari modul cacah pulsa yang menggunakan metode interrupt dan dari modul cacah yang menggunakan metode timer/counter. Skenario pencuplikan data dilakukan sebagai berikut;

- 1) Mesin pembangkit cacah pulsa; dikenal secara umum dengan sebutan pulse generator; diatur sedemikian rupa sehingga menghasilkan nilai cacah dengan jumlah cacah pulsa perdetik pada nilai tertentu.
- 2) Nilai cacah tersebut dilewatkan ke port masukan dari kedua modul cacah (mode interupsi dan mode timer/counter) berbasis mikrokontroler STM32F103C8T6 tersebut.
- 3) Nilai cacah pulsa hasil pembacaan kedua modul tersebut dimasukkan ke dalam tabel yang telah disediakan.
- 4) Pengaturan nilai cacah pada generator pulsa dilakukan secara berulang dan diurutkan mulai dari nilai cacah pulsa bernilai rendah hingga bernilai tinggi.
- 5) Setelah proses pengambilan data hasil cacah selesai, kedua hasil cacah pulsa dari mode interupsi dan mode timer/counter tersebut dibandingkan dengan hasil cacah yang tertulis pada mesin pembangkit pulsa. Model cacah yang memiliki nilai hasil komparasi yang paling mendekati hasil cacah yang tercatat pada mesin generator pulsa dianggap dan disimpulkan sebagai model cacah yang paling baik. Bentuk tabulasi hasil cacah dari mode interupsi dan mode Timer/Counter diperlihatkan oleh Gambar 4.2.

No	Pulse Generator Cacah/detik	Hasil Cacah Mode Interrupt	Hasil Cacah Mode Timer/Counter
1			
2			
3			
4			
.			
.			
N			

Gambar 4.2. Bentuk Tabel Tabulasi Hasil Cacah

4.2. Alat dan Bahan

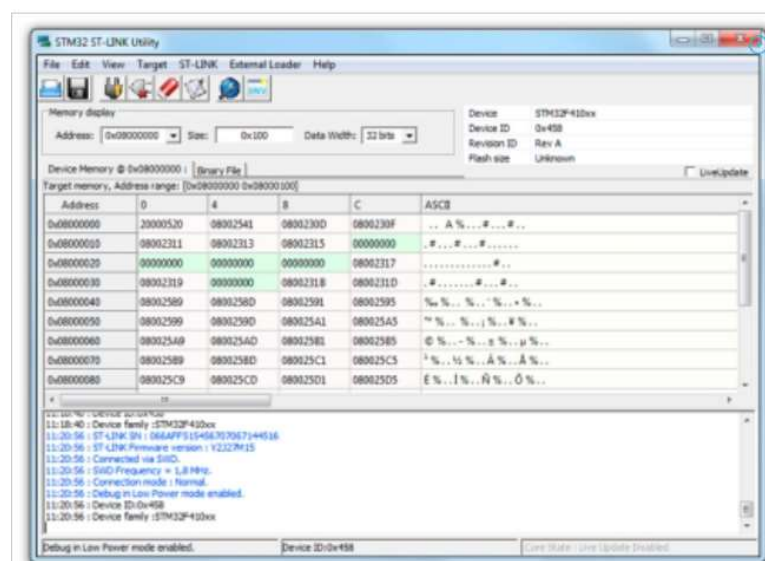
Alat yang digunakan dalam penelitian ini adalah antara lain:

1. Komputer Laptop (1 buah)

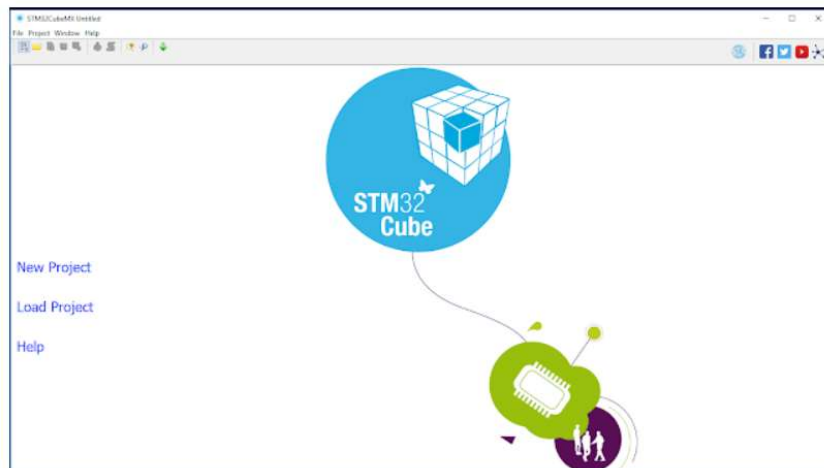
Komputer laptop (Gambar 4.3) digunakan untuk membuat program, mengkonversi source code ke dalam bentuk file hex, dan mengunggah ke dalam mikrokontroler STM32F103C8T6 melalui perantara perangkat lunak STM32 ST-LINK Utility (STSW-LINK004) (Gambar 4.4). Untuk membuat program yang akan ditanamkan ke mikrokontroler digunakan dua buah perangkat lunak aplikasi, yaitu STM32CubeMX (Gambar 4.5) dan Atollic TrueSTUDIO untuk ARM (Gambar 4.6). Seluruhnya versi Windows.



Gambar 4.3. Laptop untuk membuat program dan mengunggah file hex.



Gambar 4.4. STM32 ST-LINK Utility (STSW-LINK004)



Gambar 4.5. STM32CubeMX



Gambar 4.6. Atollic TrueSTUDIO untuk ARM

2. ST-Link V2

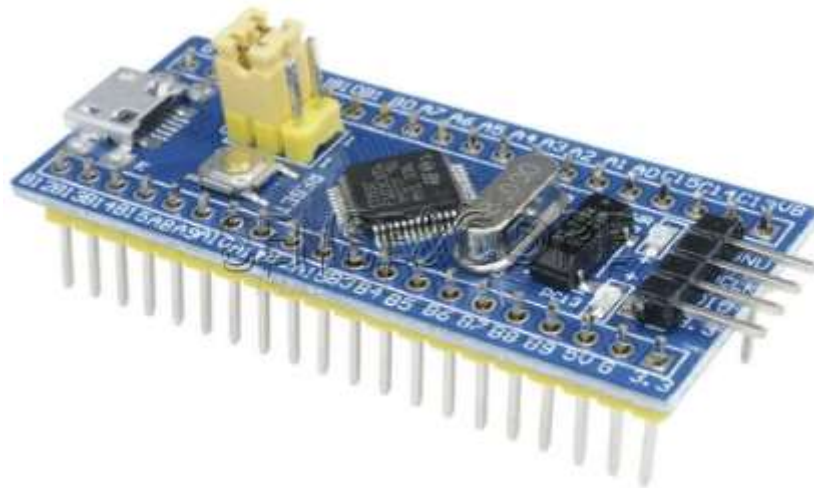
ST-Link V2 digunakan sebagai antarmuka komunikasi untuk mengunggah program dalam bentuk kode hex ke dalam mikrokontroler (Gambar 4.7)



Gambar 4.7 ST-Link V2

3. Mikrokontroler STM32F103C8T6

STM32F103C8T6 adalah mikrokontroler versi ARM yang akan digunakan untuk melakukan proses pencacahan pulsa dengan teknik interrupt dan timer/counter (Gambar 4.8).



Gambar 4.8. Mikrokontroler STM32F103C8T6

4. LCD 20x4

LCD 20x4 (Gambar 4.9) dengan driver I2C diperlukan untuk menampilkan informasi hasil cacah pulsa persatuan waktu



Gambar 4.9. LCD 20x4 dengan driver I2C.

5. Kabel data micro USB

Kabel data micro USB (Gambar 4.10) digunakan untuk memberi daya mikrokontroler dari Laptop.



Gambar 4.10. Kabel data/daya micro USB

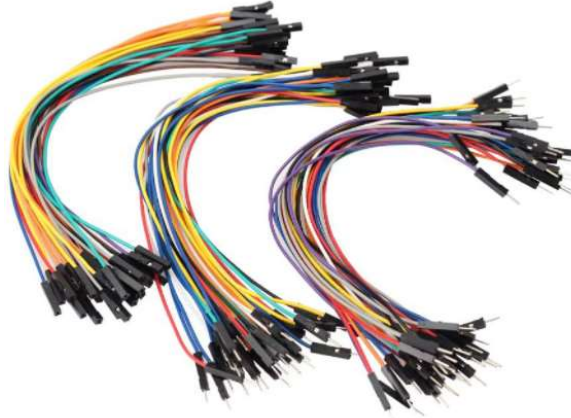
6. Pulse generator

Pulse Generator (Gambar 4.11) merupakan alat yang digunakan untuk menghasilkan jumlah pulsa/event per satuan detik.



Gambar 4.11. Pulse Generator

Bahan yang digunakan untuk membantu melancarkan jalannya penelitian adalah kabel jumper. Kabel jumper digunakan untuk menghubungkan antara dua konektor yang terpisah (Gambar 4.12). Ada beberapa jenis kabel jumper yang sering dijumpai di toko-toko elektronika, yaitu jumper male to male, male to female, dan female to female.



Gambar 4.12. Contoh kabel jumper

4.3. Langkah-langkah Penelitian

Secara umum, langkah-langkah penelitian yang dilakukan untuk membuat modul mikrokontroler sebagai pencacah pulsa berbasis interrupt dan timer/counter adalah sebagai berikut:

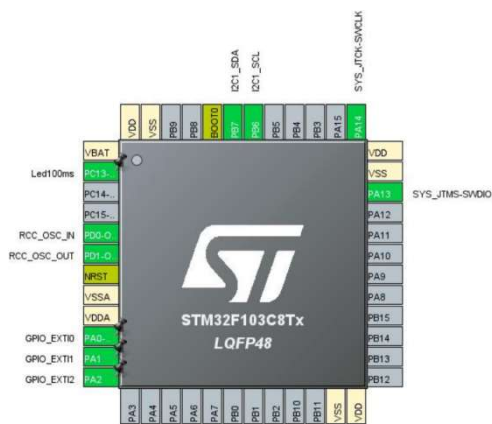
1. Melakukan konfigurasi mikrokontroler untuk pencacah pulsa berbasis interrupt dan timer/counter menggunakan perangkat lunak konfigurator, yaitu STM32CubeMX.
2. Mengkonversi data hasil konfigurasi tersebut ke dalam bahasa C/C++ untuk diuraikan lebih lanjut hingga menghasilkan program pencacah pulsa pada mikrokontroler dengan mode interrupt dan mode timer/counter
3. Mengkompilasi program tersebut ke dalam format file hex.
4. Mengunggah file hex ke dalam mikrokontroler.
5. Menguji mikrokontroler yang telah berisi file hex mode interupsi dan mode timer/counter dengan memberikan pulsa cacah pada bagian input pulsa dari mikrokontroler. Kemudian melakukan pencatatan atas hasil yang ditampilkan pada LCD 20x4.

BAB V

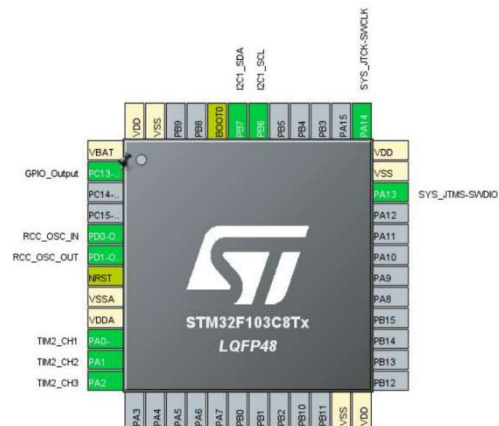
HASIL DAN PEMBAHASAN

5.1. Hasil Konfigurasi Mikrokontroler

Bentuk fisik hasil konfigurasi mikrokontroler menggunakan metode intrupsi dan metode timer counter pada STM32CubeMX secara berurutan diperlihatkan oleh Gambar 5.1 dan Gambar 5.2.



Gambar 5.1.
Konfigurasi mode intrupsi



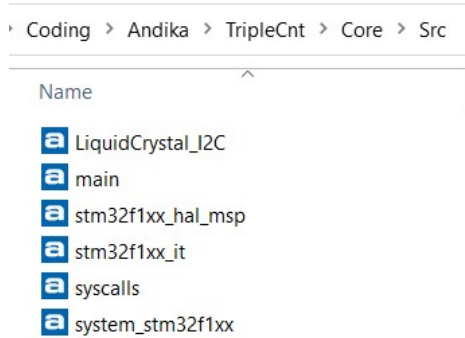
Gambar 5.2
Konfigurasi mode timer/counter

Perbedaan utama konfigurasi mikrokontroler antara yang menggunakan metode interrupt dan metode timer/counter secara fisik terlihat pada pin masukan pulsa cacah. Untuk metode interrupt pin masukannya diset sebagai kanal interrupt GPIO. Sementara untuk metode timer/counter pin masukannya diset sebagai kanal timer. Bentuk konfigurasi secara detail dapat dilihat pada Lampiran 1 dan 2 dari laporan penelitian ini.

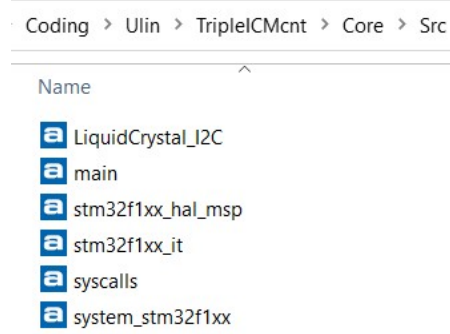
5.2. Hasil Implementasi Program

Hasil implementasi program untuk menampilkan nilai cacah pulsa dari metode interrupt dan metode timer/counter secara umum dipecah ke dalam 5 file (Gambar 5.3 dan 5.4). Dari ke-5 file tersebut ada 2 file yang memegang peran utama, yaitu file main.c dan file stm32f1xx_it.c. File main.c berisi program untuk memanggil dan menjalankan program yang ada di dalam file stm32f1xx_it.c serta

menampilkan hasilnya pada layer LCD 20x4. Sementara file `stm32f1xx_it.c` adalah file yang digunakan untuk mengimplementasikan metode cacah pulsa, baik cacah pulsa yang menggunakan metode interrupt maupun dengan metode timer/counter.



Gambar 5.3.
File-file program interrupt



Gambar 5.4.
File-file program timer/counter

5.3. Analisis Hasil Komparasi

Data yang diperoleh dari pengujian masing-masing modul menggunakan pulse generator disajikan dalam bentuk tabulasi sebagaimana diperlihatkan dalam Tabel 2.

Tabel 2. Data Hasil Percobaan

No	Pls. Gen	Cacah Intr.	Cacah Tmr/Cnt
1	100	100	100
2	500	502	503
3	1000	1002	998
4	5000	5002	5005
5	7500	7502	7504
6	10000	10004	9997
7	20000	2003	2002
8	30000	30002	29997
9	40000	40003	40004
10	50000	50002	50002
11	60000	60003	59997
12	70000	70004	70004
13	80000	80002	80003
14	90000	90003	90002
15	100000	10004	10005

Berdasarkan hasil uji beda menggunakan uji t, diketahui bahwa hasil cacah yang diperoleh dari mode interrupt dan mode timer/counter tidak memiliki perbedaan yang nyata terhadap cacah pulsa yang dihasilkan oleh pembangkit pulsa pada tingkat nilai kepercayaan sebesar 5%.

Uji beda menggunakan uji t juga dilakukan terhadap hasil cacah antara mode interrupt dan mode timer/counter. Dari hasil uji beda tersebut, keduanya menunjukkan tidak memiliki perbedaan yang signifikan pada tingkat kepercayaan 5%.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari uji coba dan pengolahan data yang telah dilakukan, penulis berkesimpulan bahwa tidak ada perbedaan nyata antara pencacah pulsa yang menggunakan metode interrupt dan metode timer/counter.

Pencacah pulsa berbasis mikrokontroler STM32F103C8T6 dengan kecepatan kejadian dari 1 detik perkejadian hingga 100000 kejadian perdetik dapat menggunakan metode interrupt maupun metode timer/counter.

6.2. Saran

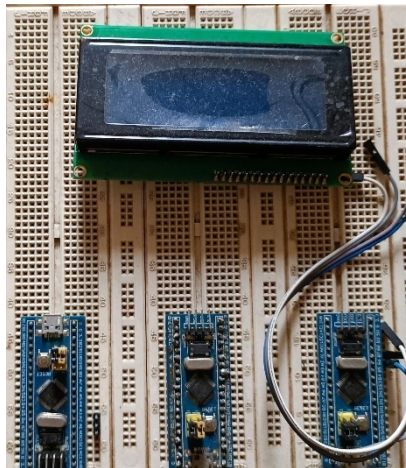
Masih terjadi simpangan-simpangan kecil pada sejumlah hasil cacah pulsa. Simpangan-simpangan tersebut dapat terjadi karena sumber cacah sendiri yang kurang stabil, atau timer untuk menentukan waktu cacah perdetik yang bermasalah karena adanya pembulatan perhitungan. Oleh karena itu, untuk memastikan sumber penyebab dari simpangan-simpangan kecil, diperlukan sumber cacah yang presisi dan juga sumber timer yang stabil pada mikrokontroler.

DAFTAR PUSTAKA

- [1] Anonim, *Materi Panduan Praktikum Elektronika Dasar, Laboratorium Elektronika dan Komputer Universitas Gunadarma*, Depok, 2008
- [2] Bejo, Agus. 2008. *C & AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATMEGA8535*. Yogyakarta, Ghara Ilmu.
- [3] Widodo Budiharto, 2008 Sigit Firmansyah, *Elektronika Digital dan Mikroprosesor*, Andi, Yogyakarta.
- [4] URL: <http://tutorial-elektronika.blogspot.com/2009/02/ic-lm-324.html>
- [5] URL: <http://www.datasheet.com/lm234.html>, 2008
- [6] URL: <http://elektronika-dasar.com/teori-elektronika/teori-motor-dc-dan-jenis-jenis-motor-dc/>
- [7] URL: <http://meriwardana.blogspot.com/2011/11/prinsip-kerja-motor-arus-searah-dc.html>.

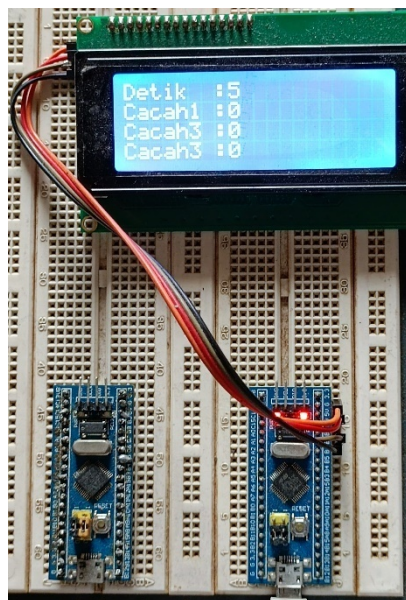
LAMPIRAN 1

Tampilan sistem pencacah pulsa berbasis STM32F103C8T6 dan LCD 20x4
(Gambar L1.1)



Gambar L1.1. Pencacah Pulsa berbasis STM32F103C8T6

Hasil uji koneksi antara mikrokontroler dan LCD via I2C (Gambar L1.2)



Gambar L1.2. Uji Koneksi Antara LCD dan STM32F103C8T6

LAMPIRAN 2

Konfigurasi STM32F103C8T6 untuk mode Cacah Pulsa Interrupt

TripleCnt Project
Configuration Report

1. Description

1.1. Project

Project Name	TripleCnt
Board Name	custom
Generated with:	STM32CubeMX 6.3.0
Date	08/14/2021

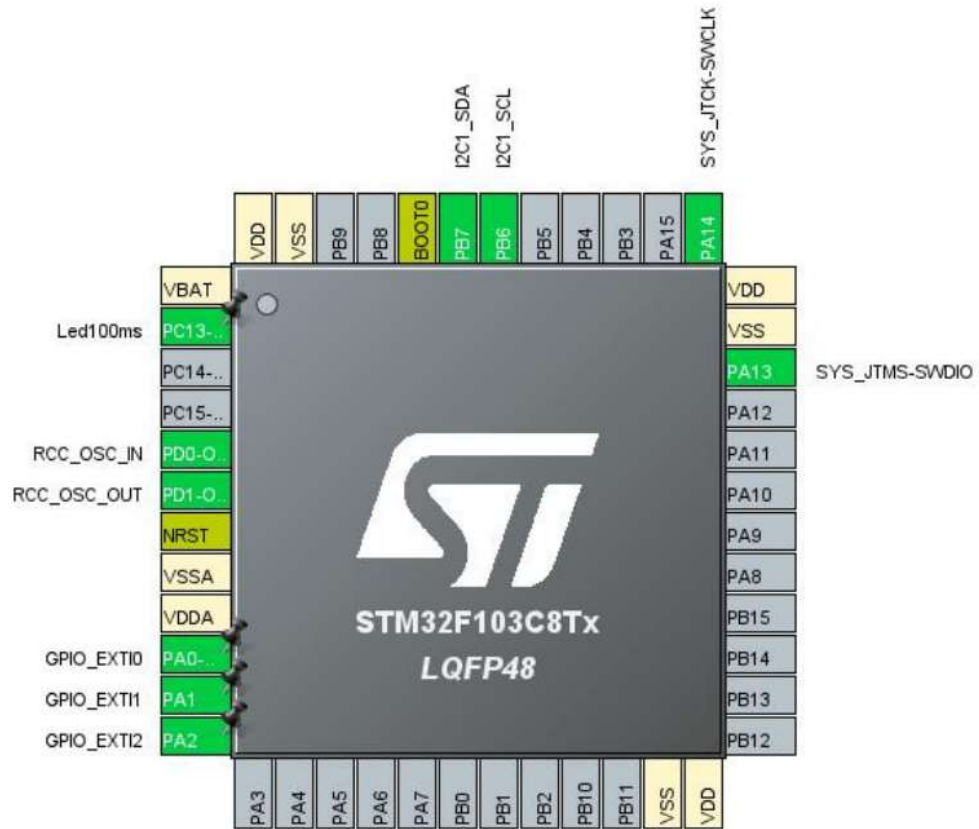
1.2. MCU

MCU Series	STM32F1
MCU Line	STM32F103
MCU name	STM32F103C8Tx
MCU Package	LQFP48
MCU Pin number	48

1.3. Core(s) information

Core(s)	Arm Cortex-M3
---------	---------------

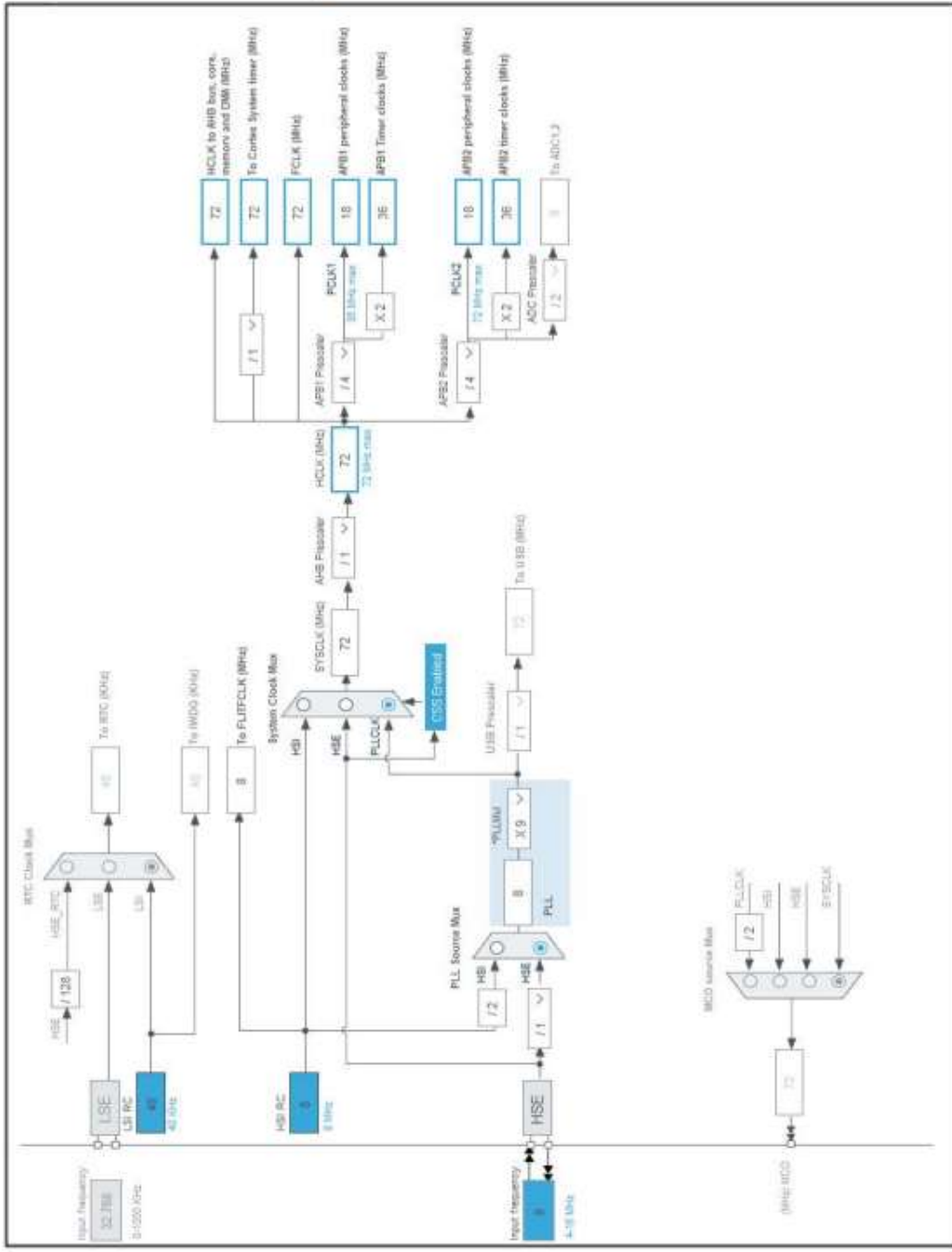
2. Pinout Configuration



3. Pins Configuration

Pin Number LQFP48	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
1	VBAT	Power		
2	PC13-TAMPER-RTC *	I/O	GPIO Output	Led100ms
5	PD0-OSC_IN	I/O	RCC_OSC_IN	
6	PD1-OSC_OUT	I/O	RCC_OSC_OUT	
7	NRST	Reset		
8	VSSA	Power		
9	VDDA	Power		
10	PA0-WKUP	I/O	GPIO_EXTI0	
11	PA1	I/O	GPIO_EXTI1	
12	PA2	I/O	GPIO_EXTI2	
23	VSS	Power		
24	VDD	Power		
34	PA13	I/O	SYS_JTMS-SWDIO	
35	VSS	Power		
36	VDD	Power		
37	PA14	I/O	SYS_JTCK-SWCLK	
42	PB6	I/O	I2C1_SCL	
43	PB7	I/O	I2C1_SDA	
44	BOOT0	Boot		
47	VSS	Power		
48	VDD	Power		

4. Clock Tree Configuration



5. Software Project

5.1. Project Settings

Name	Value
Project Name	TripleCnt
Project Folder	D:\PKL07_21\TripleCnt
Toolchain / IDE	TrueSTUDIO
Firmware Package Name and Version	STM32Cube FW_F1 V1.8.4
Application Structure	Advanced
Generate Under Root	Yes
Do not generate the main()	No
Minimum Heap Size	0x200
Minimum Stack Size	0x400

5.2. Code Generation Settings

Name	Value
STM32Cube MCU packages and embedded software	Copy only the necessary library files
Generate peripheral initialization as a pair of '.c/.h' files	No
Backup previously generated files when re-generating	No
Keep User Code when re-generating	Yes
Delete previously generated files when not re-generated	Yes
Set all free pins as analog (to optimize the power consumption)	Yes
Enable Full Assert	No

5.3. Advanced Settings - Generated Function Calls

Rank	Function Name	Peripheral Instance Name
1	MX_GPIO_Init	GPIO
2	SystemClock_Config	RCC
3	MX_I2C1_Init	I2C1
4	MX_TIM2_Init	TIM2

6. Power Consumption Calculator report

6.1. Microcontroller Selection

Series	STM32F1
Line	STM32F103
MCU	STM32F103C8Tx
Datasheet	DS5319 Rev17

6.2. Parameter Selection

Temperature	25
Vdd	3.3

6.3. Battery Selection

Battery	Li-SOCL2(A3400)
Capacity	3400.0 mAh
Self Discharge	0.08 %/month
Nominal Voltage	3.6 V
Max Cont Current	100.0 mA
Max Pulse Current	200.0 mA
Cells in series	1
Cells in parallel	1

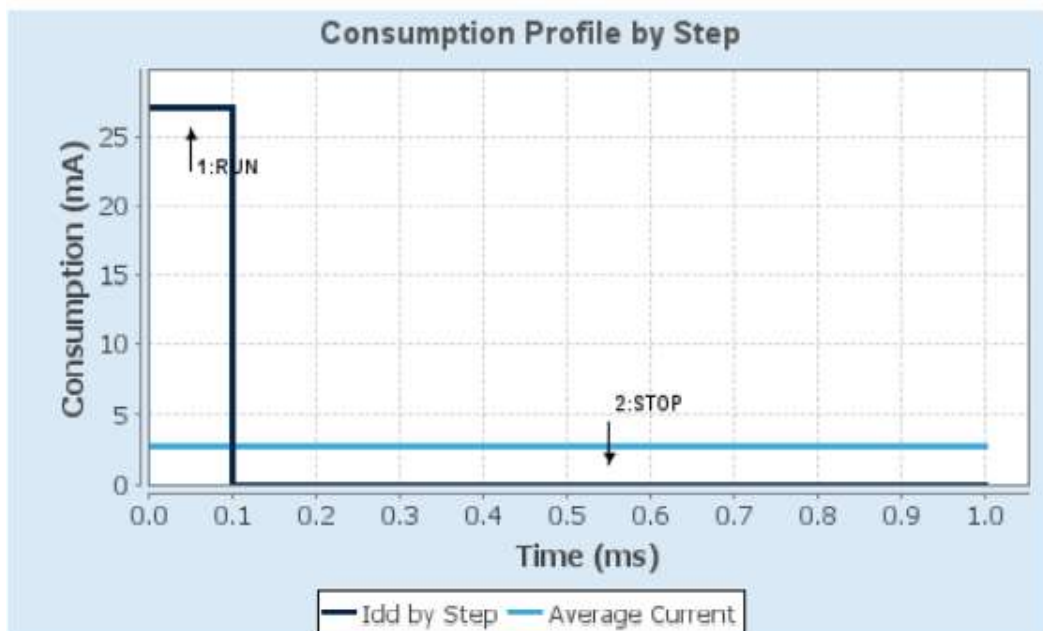
6.4. Sequence

Step	Step1	Step2
Mode	RUN	STOP
Vdd	3.3	3.3
Voltage Source	Battery	Battery
Range	No Scale	No Scale
Fetch Type	FLASH	n/a
CPU Frequency	72 MHz	0 Hz
Clock Configuration	HSE PLL	Regulator_LP
Clock Source Frequency	8 MHz	0 Hz
Peripherals		
Additional Cons.	0 mA	0 mA
Average Current	27 mA	14 μ A
Duration	0.1 ms	0.9 ms
DMIPS	90.0	0.0
Ta Max	100.1	105
Category	In DS Table	In DS Table

6.5. Results

Sequence Time	1 ms	Average Current	2.71 mA
Battery Life	1 month, 21 days, 17 hours	Average DMIPS	61.0 DMIPS

6.6. Chart



7. Peripherals and Middlewares Configuration

7.1. I2C1

I2C: I2C

7.1.1. Parameter Settings:

Master Features:

I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000

Slave Features:

Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

7.2. RCC

High Speed Clock (HSE): Crystal/Ceramic Resonator

7.2.1. Parameter Settings:

System Parameters:

VDD voltage (V)	3.3
Prefetch Buffer	Enabled
Flash Latency(WS)	2 WS (3 CPU cycle)

RCC Parameters:

HSI Calibration Value	16
HSE Startup Timeout Value (ms)	100
LSE Startup Timeout Value (ms)	5000

7.3. SYS

Debug: Serial Wire

Timebase Source: SysTick

7.4. TIM2

Clock Source : Internal Clock

7.4.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	36000-1 *
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	2000 *
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

Trigger Output (TRGO) Parameters:

Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

* User modified value

8. System Configuration

8.1. GPIO configuration

IP	Pin	Signal	GPIO mode	GPIO pull/up pull down	Max Speed	User Label
I2C1	PB6	I2C1_SCL	Alternate Function Open Drain	n/a	High *	
	PB7	I2C1_SDA	Alternate Function Open Drain	n/a	High *	
RCC	PD0-OSC_IN	RCC_OSC_IN	n/a	n/a	n/a	
	PD1-OSC_OUT	RCC_OSC_OUT	n/a	n/a	n/a	
SYS	PA13	SYS_JTMS-SWDIO	n/a	n/a	n/a	
	PA14	SYS_JTCK-SWCLK	n/a	n/a	n/a	
GPIO	PC13-TAMPFR-RTC	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Medium *	Led100ms
	PA0-WKUP	GPIO_EXTI0	External Interrupt Mode with Rising edge trigger detection	Pull-down *	n/a	
	PA1	GPIO_EXTI1	External Interrupt Mode with Rising edge trigger detection	Pull-down *	n/a	
	PA2	GPIO_EXTI2	External Interrupt Mode with Rising edge trigger detection	Pull-down *	n/a	

8.2. DMA configuration

nothing configured in DMA service

8.3. NVIC configuration

8.3.1. NVIC

Interrupt Table	Enable	Preenmption Priority	SubPriority
Non maskable interrupt	true	0	0
Hard fault interrupt	true	0	0
Memory management fault	true	0	0
Prefetch fault, memory access fault	true	0	0
Undefined instruction or illegal state	true	0	0
System service call via SWI instruction	true	0	0
Debug monitor	true	0	0
Pendable request for system service	true	0	0
System tick timer	true	16	0
RCC global interrupt	true	0	0
EXTI line0 interrupt	true	0	0
EXTI line1 interrupt	true	0	0
EXTI line2 interrupt	true	0	0
TIM2 global interrupt	true	0	0
PVD interrupt through EXTI line 16		unused	
Flash global interrupt		unused	
I2C1 event interrupt		unused	
I2C1 error interrupt		unused	

8.3.2. NVIC Code generation

Enabled interrupt Table	Select for init sequence ordering	Generate IRQ handler	Call HAL handler
Non maskable interrupt	false	true	true
Hard fault interrupt	false	true	false
Memory management fault	false	true	false
Prefetch fault, memory access fault	false	true	false
Undefined instruction or illegal state	false	true	false
System service call via SWI instruction	false	true	false
Debug monitor	false	true	false
Pendable request for system service	false	true	false
System tick timer	false	true	true
RCC global interrupt	false	true	false
EXTI line0 interrupt	false	true	true
EXTI line1 interrupt	false	true	true
EXTI line2 interrupt	false	true	true
TIM2 global interrupt	false	true	true

9. System Views

9.1. Category view

9.1.1. Current



LAMPIRAN 3

Konfigurasi STM32F103C8T6 untuk mode cacah pulsa Timer/Counter

1. Description

1.1. Project

Project Name	TripleICMcnt
Board Name	custom
Generated with:	STM32CubeMX 6.3.0
Date	08/17/2021

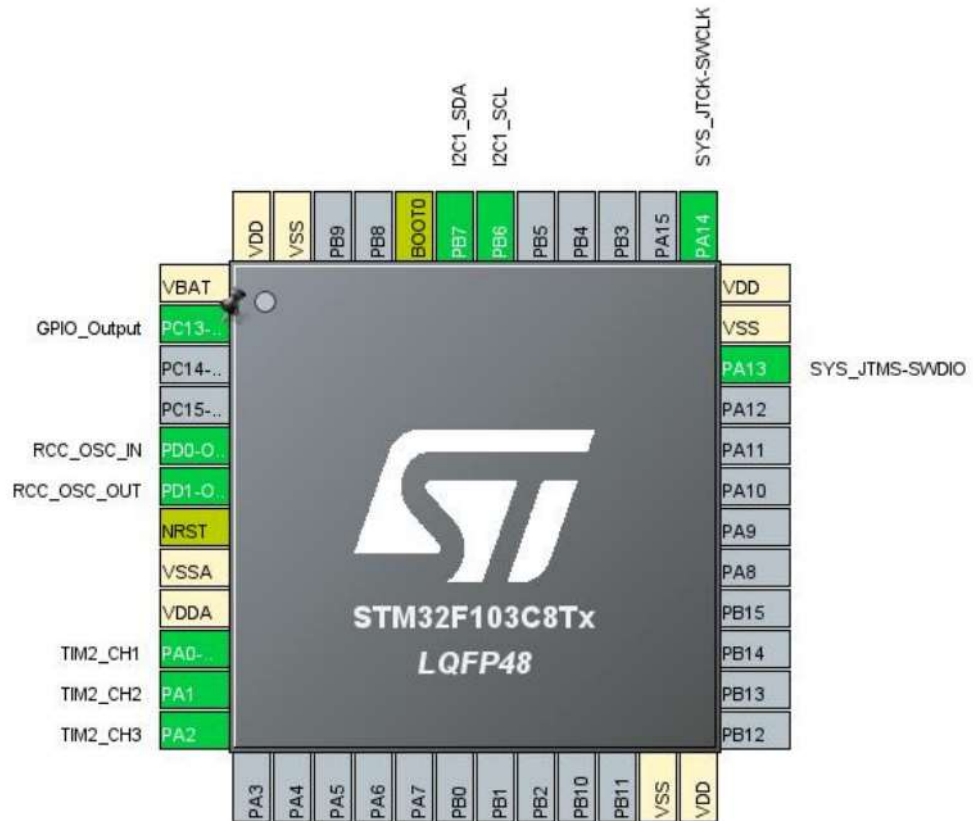
1.2. MCU

MCU Series	STM32F1
MCU Line	STM32F103
MCU name	STM32F103C8Tx
MCU Package	LQFP48
MCU Pin number	48

1.3. Core(s) information

Core(s)	Arm Cortex-M3
---------	---------------

2. Pinout Configuration



3. Pins Configuration

Pin Number LQFP48	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
1	VBAT	Power		
2	PC13-TAMPER-RTC *	I/O	GPIO Output	
5	PD0-OSC IN	I/O	RCC OSC IN	
6	PD1-OSC OUT	I/O	RCC OSC OUT	
7	NRST	Reset		
8	VSSA	Power		
9	VDDA	Power		
10	PA0-WKUP	I/O	TIM2 CH1	
11	PA1	I/O	TIM2 CH2	
12	PA2	I/O	TIM2 CH3	
23	VSS	Power		
24	VDD	Power		
34	PA13	I/O	SYS_JTMS-SWDIO	
35	VSS	Power		
36	VDD	Power		
37	PA14	I/O	SYS_JTCK-SWCLK	
42	PB6	I/O	I2C1_SCL	
43	PB7	I/O	I2C1_SDA	
44	BOOT0	Boot		
47	VSS	Power		
48	VDD	Power		

5. Software Project

5.1. Project Settings

Name	Value
Project Name	TriplelCMcnt
Project Folder	D:\PKL07_21\TriplelCMcnt
Toolchain / IDE	TrueSTUDIO
Firmware Package Name and Version	STM32Cube FW_F1 V1.8.4
Application Structure	Advanced
Generate Under Root	Yes
Do not generate the main()	No
Minimum Heap Size	0x200
Minimum Stack Size	0x400

5.2. Code Generation Settings

Name	Value
STM32Cube MCU packages and embedded software	Copy only the necessary library files
Generate peripheral initialization as a pair of '.c/.h' files	No
Backup previously generated files when re-generating	No
Keep User Code when re-generating	Yes
Delete previously generated files when not re-generated	Yes
Set all free pins as analog (to optimize the power consumption)	Yes
Enable Full Assert	No

5.3. Advanced Settings - Generated Function Calls

Rank	Function Name	Peripheral Instance Name
1	MX_GPIO_Init	GPIO
2	SystemClock_Config	RCC
3	MX_I2C1_Init	I2C1
4	MX_TIM3_Init	TIM3
5	MX_DMA_Init	DMA
6	MX_TIM2_Init	TIM2

6. Power Consumption Calculator report

6.1. Microcontroller Selection

Series	STM32F1
Line	STM32F103
MCU	STM32F103C8Tx
Datasheet	DS5319_Rev17

6.2. Parameter Selection

Temperature	25
Vdd	3.3

6.3. Battery Selection

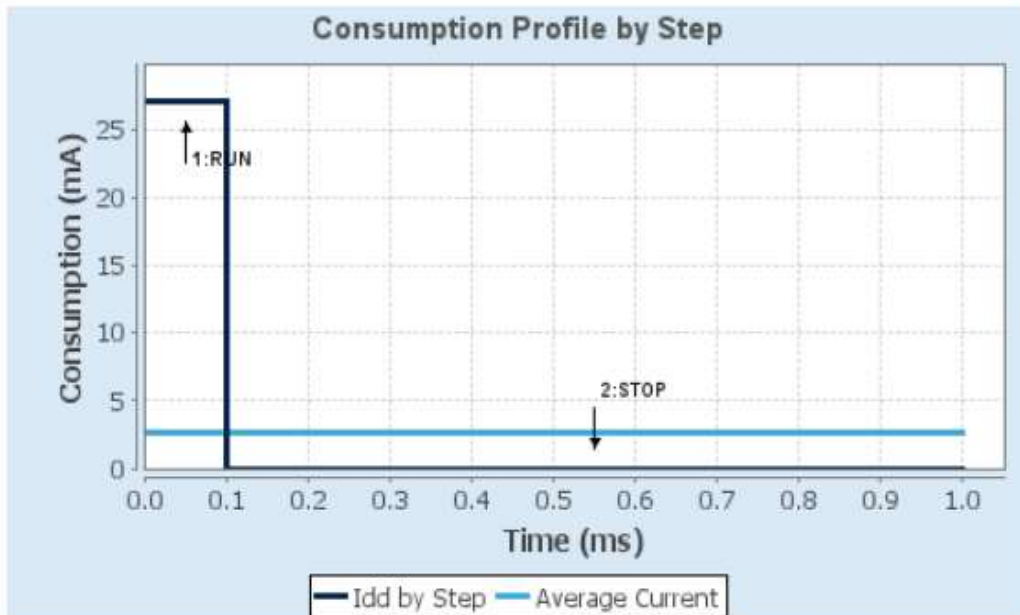
Battery	Li-SOCL2(A3400)
Capacity	3400.0 mAh
Self Discharge	0.08 %/month
Nominal Voltage	3.6 V
Max Cont Current	100.0 mA
Max Pulse Current	200.0 mA
Cells in series	1
Cells in parallel	1

6.4. Sequence

Step	Step1	Step2
Mode	RUN	STOP
Vdd	3.3	3.3
Voltage Source	Battery	Battery
Range	No Scale	No Scale
Fetch Type	FLASH	n/a
CPU Frequency	72 MHz	0 Hz
Clock Configuration	HSE PLL	Regulator LP
Clock Source Frequency	8 MHz	0 Hz
Peripherals		
Additional Cons.	0 mA	0 mA
Average Current	27 mA	14 μ A
Duration	0.1 ms	0.9 ms
DMIPS	90.0	0.0
Ta Max	100.1	105
Category	In DS Table	In DS Table

6.5. Results

Sequence Time	1 ms	Average Current	2.71 mA
Battery Life	1 month, 21 days, 17 hours	Average DMIPS	61.0 DMIPS



7. Peripherals and Middlewares Configuration

7.1. I2C1

I2C: I2C

7.1.1. Parameter Settings:

Master Features:

I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000

Slave Features:

Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

7.2. RCC

High Speed Clock (HSE): Crystal/Ceramic Resonator

7.2.1. Parameter Settings:

System Parameters:

VDD voltage (V)	3.3
Prefetch Buffer	Enabled
Flash Latency(WS)	1 WS (2 CPU cycle)

RCC Parameters:

HSI Calibration Value	16
HSE Startup Timeout Value (ms)	100
LSE Startup Timeout Value (ms)	5000

7.3. SYS

Debug: Serial Wire

Timebase Source: SysTick

7.4. TIM2

Clock Source : Internal Clock

Channel1: Input Capture direct mode

Channel2: Input Capture direct mode

Channel3: Input Capture direct mode

7.4.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	180-1 *
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	50000 *
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

Trigger Output (TRGO) Parameters:

Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Input Capture Channel 1:

Polarity Selection	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter (4 bits value)	15 *

Input Capture Channel 2:

Polarity Selection	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter (4 bits value)	15 *

Input Capture Channel 3:

Polarity Selection	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter (4 bits value)	15 *

7.5. TIM3

mode: Clock Source

7.5.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	36000-1 *
Counter Mode	Up

8. System Configuration

8.1. GPIO configuration

IP	Pin	Signal	GPIO mode	GPIO pull/up pull down	Max Speed	User Label
I2C1	PB6	I2C1_SCL	Alternate Function Open Drain	n/a	High *	
	PB7	I2C1_SDA	Alternate Function Open Drain	n/a	High *	
RCC	PD0-OSC_IN	RCC_OSC_IN	n/a	n/a	n/a	
	PD1-OSC_OUT	RCC_OSC_OUT	n/a	n/a	n/a	
SYS	PA13	SYS_JTMS-SWDIO	n/a	n/a	n/a	
	PA14	SYS_JTCK-SWCLK	n/a	n/a	n/a	
TIM2	PA0-WKUP	TIM2_CH1	Input mode	Pull-down *	n/a	
	PA1	TIM2_CH2	Input mode	Pull-down *	n/a	
	PA2	TIM2_CH3	Input mode	Pull-down *	n/a	
GPIO	PC13-TAMPER-RTC	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Medium *	

8.2. DMA configuration

DMA request	Stream	Direction	Priority
TIM2_CH1	DMA1_Channel5	Peripheral To Memory	Low
TIM2_CH2/CH4	DMA1_Channel7	Peripheral To Memory	Low
TIM2_CH3	DMA1_Channel1	Peripheral To Memory	Low

TIM2_CH1: DMA1_Channel5 DMA request Settings:

Mode: **Circular ***
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: **Word ***
Memory Data Width: **Word ***

TIM2_CH2/CH4: DMA1_Channel7 DMA request Settings:

Mode: **Circular ***
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: **Word ***
Memory Data Width: **Word ***

TIM2_CH3: DMA1_Channel1 DMA request Settings:

Mode: **Circular ***
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: **Word ***
Memory Data Width: **Word ***

8.3. NVIC configuration

8.3.1. NVIC

Interrupt Table	Enable	Preenmption Priority	SubPriority
Non maskable interrupt	true	0	0
Hard fault interrupt	true	0	0
Memory management fault	true	0	0
Prefetch fault, memory access fault	true	0	0
Undefined instruction or illegal state	true	0	0
System service call via SWI instruction	true	0	0
Debug monitor	true	0	0
Pendable request for system service	true	0	0
System tick timer	true	15	0
RCC global interrupt	true	0	0
DMA1 channel1 global interrupt	true	0	0
DMA1 channel5 global interrupt	true	0	0
DMA1 channel7 global interrupt	true	0	0
TIM2 global interrupt	true	0	0
TIM3 global interrupt	true	15	0
PVD interrupt through EXTI line 16		unused	
Flash global interrupt		unused	
I2C1 event interrupt		unused	
I2C1 error interrupt		unused	

8.3.2. NVIC Code generation

Enabled interrupt Table	Select for init sequence ordering	Generate IRQ handler	Call HAL handler
Non maskable interrupt	false	true	true
Hard fault interrupt	false	true	false
Memory management fault	false	true	false
Prefetch fault, memory access fault	false	true	false
Undefined instruction or illegal state	false	true	false
System service call via SWI instruction	false	true	false
Debug monitor	false	true	false
Pendable request for system service	false	true	false
System tick timer	false	true	true
RCC global interrupt	false	true	false
DMA1 channel1 global interrupt	false	true	true
DMA1 channel5 global interrupt	false	true	true
DMA1 channel7 global interrupt	false	true	true
TIM2 global interrupt	false	true	true

9. System Views

9.1. Category view

9.1.1. Current



LAMPIRAN 4

L4.1 Potongan Program Utama Mode Interrupt

```
lap1.cpp  I2C_LCD.c  main.c ✖
127  MX_GPIO_Init();
128  MX_I2C1_Init();
129  MX_TIM2_Init();
130  MX_TIM3_Init();
131  /* USER CODE BEGIN 2 */
132  HAL_TIM_Base_Start_IT(&htim3);
133  HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_1);
134  HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_2);
135  HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_3);
136  //-----
137  LiquidCrystal_I2C_Init(&hi2c1, 0x3F, 20, 4, LCD_5x8DOTS);
138  LiquidCrystal_I2C_Clear();
139  LiquidCrystal_I2C_SetBacklight(1);
140  LiquidCrystal_I2C_SetCursor(0,0);
141  LiquidCrystal_I2C_PrintString("*****");
142  LiquidCrystal_I2C_SetCursor(0,1);
143  LiquidCrystal_I2C_PrintString("** TIMER/COUNTER **");
144  LiquidCrystal_I2C_SetCursor(0,2);
145  LiquidCrystal_I2C_PrintString("*** By M. Ulin N.***");
146  LiquidCrystal_I2C_SetCursor(0,3);
147  LiquidCrystal_I2C_PrintString("*****");
148
```

L4.2 Potongan Program Utama Mode Timer/Counter

```
lap1.cpp  I2C_LCD.c  *main.c  *main.c ✖  stm32f1xx_it.c  str
106  /* USER CODE BEGIN 2 */
107  //-----
108  HAL_TIM_Base_Start_IT(&htim2); //start interupsi timer untk
109  //-----
110  LiquidCrystal_I2C_Init(&hi2c1, 0x3F, 20, 4, LCD_5x8DOTS);
111  LiquidCrystal_I2C_Clear();
112  LiquidCrystal_I2C_SetBacklight(1);
113  LiquidCrystal_I2C_SetCursor(0,0);
114  LiquidCrystal_I2C_PrintString("*****");
115  LiquidCrystal_I2C_SetCursor(0,1);
116  LiquidCrystal_I2C_PrintString("CNT. INTERRUPT MODE");
117  LiquidCrystal_I2C_SetCursor(0,2);
118  LiquidCrystal_I2C_PrintString("*** By Andika F.***");
119  LiquidCrystal_I2C_SetCursor(0,0);
120  LiquidCrystal_I2C_PrintString("*****");
121  HAL_Delay(5000); //tunggu 5 detik sebelum beroperasi
122  LiquidCrystal_I2C_Clear();
123  /* USER CODE END 2 */
```

L4.3. Potongan Program Mode Cacah Pulsa Interrupt

```
lap1.cpp I2C_LCD.c *main.c *main.c stm32f1xx_it.c stm32f1xx_it.c
227 */
228 void EXTI0_IRQHandler(void)
229 {
230     /* USER CODE BEGIN EXTI0_IRQn 0 */
231     cch1 = cch1+1;
232     /* USER CODE END EXTI0_IRQn 0 */
233     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
234     /* USER CODE BEGIN EXTI0_IRQn 1 */
235
236     /* USER CODE END EXTI0_IRQn 1 */
237 }
238
239 /**
240  * @brief This function handles EXTI line1 interrupt.
241  */
242 void EXTI1_IRQHandler(void)
243 {
244     /* USER CODE BEGIN EXTI1_IRQn 0 */
245     cch2 = cch2+1;
246     /* USER CODE END EXTI1_IRQn 0 */
247     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1);
248     /* USER CODE BEGIN EXTI1_IRQn 1 */
249
250     /* USER CODE END EXTI1_IRQn 1 */
251 }
252
253 /**
254  * @brief This function handles EXTI line2 interrupt.
255  */
256 void EXTI2_IRQHandler(void)
257 {
258     /* USER CODE BEGIN EXTI2_IRQn 0 */
259     cch3 = cch3+1;
260     /* USER CODE END EXTI2_IRQn 0 */
261     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_2);
262     /* USER CODE BEGIN EXTI2_IRQn 1 */
263
264     /* USER CODE END EXTI2_IRQn 1 */
265 }
```

L4.4. Potongan Program Mode Cacah Pulsa Timer/Counter

```
lap1.cpp *main.c stm32f1xx_it.c stm32f1xx_it.c stm32f1xx_it.c
223 void DMA1_Channel1_IRQHandler(void)
224 {
225     /* USER CODE BEGIN DMA1_Channel1_IRQn 0 */
226
227     /* USER CODE END DMA1_Channel1_IRQn 0 */
228     HAL_DMA_IRQHandler(&hdma_tim2_ch3);
229     /* USER CODE BEGIN DMA1_Channel1_IRQn 1 */
230
231     /* USER CODE END DMA1_Channel1_IRQn 1 */
232 }
233
234 /**
235  * @brief This function handles DMA1 channel5 global interrupt.
236  */
237 void DMA1_Channel5_IRQHandler(void)
238 {
239     /* USER CODE BEGIN DMA1_Channel5_IRQn 0 */
240
241     /* USER CODE END DMA1_Channel5_IRQn 0 */
242     HAL_DMA_IRQHandler(&hdma_tim2_ch1);
243     /* USER CODE BEGIN DMA1_Channel5_IRQn 1 */
244
245     /* USER CODE END DMA1_Channel5_IRQn 1 */
246 }
247
248 /**
249  * @brief This function handles DMA1 channel7 global interrupt.
250  */
251 void DMA1_Channel7_IRQHandler(void)
252 {
253     /* USER CODE BEGIN DMA1_Channel7_IRQn 0 */
254
255     /* USER CODE END DMA1_Channel7_IRQn 0 */
256     HAL_DMA_IRQHandler(&hdma_tim2_ch2_ch4);
257     /* USER CODE BEGIN DMA1_Channel7_IRQn 1 */
258
259     /* USER CODE END DMA1_Channel7_IRQn 1 */
260 }
```

LAMPIRAN 5

Uji Beda Hasil Cacah Pulsa terhadap sumber Cacah dan antara mode cacah pulsa
Menggunakan uji t, dan uji korelasi menggunakan teknik Pearson.

2									
3	100	100	100	t-Test: Two-Sample Assuming Equal Variances					
4	500	502	503						
5	1000	1002	998		<i>Variable 1</i>	<i>Variable 2</i>			
6	5000	5002	5005	Mean	31611,06667	31609,53333			
7	7500	7502	7504	Variance	978408115,2	978425888			
8	10000	10004	9997	Observations	15	15			
9	20000	2003	2002	Pooled Variance	978417001,6				
10	30000	3002	29997	Hypothesized Mean Differen	0				
11	40000	40003	40004	df	28		r	1	
12	50000	50002	50002	t Stat	0,000134247		r	1	
13	60000	60003	59997	P(T<=t) one-tail	0,499946919			<i>Column 1</i>	<i>Column 2</i>
14	70000	70004	70004	t Critical one-tail	1,701130934			Column 1	1
15	80000	80002	80003	P(T<=t) two-tail	0,999893838			Column 2	0,999999993
16	90000	90003	90002	t Critical two-tail	2,048407142				1
17	100000	10004	10005						
18						0,004999469			
19									
20			0,815995						
21			0,56317						
22			0,999933						
23									