**TUGAS 2 Jam Digital dengan 7 Segment dan 2 Push Button**

**Oleh : M. Rafly Arya P. (13216064)**

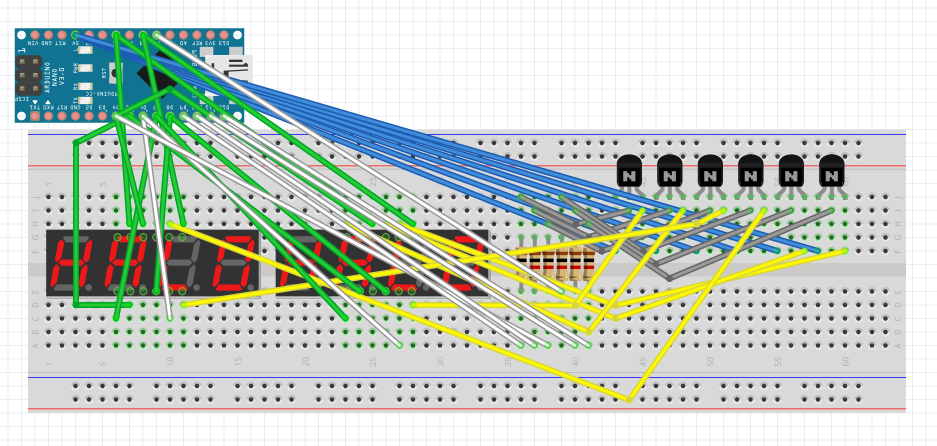
**M. Irfaan S. (13216040)**

**Gusti Triandi Winata (13216091)**

Spesifikasi

* display: 6 Digit LED 7 segment
* tidak perlu tombol
* prosesor ATmega, bisa pakai Arduino Nano ataupun Arduino UNO
* implementasi bisa breadboard, PCB lubang PCB cetak custom.
* Software menggunakan CodevisionAVR atau Atmel Studio, bukan Arduino. Setup register dilakukan manual.
* kelompok max 2 orang

Skema hardware



Penjelasan skema hardware, perhitungan nilai komponen

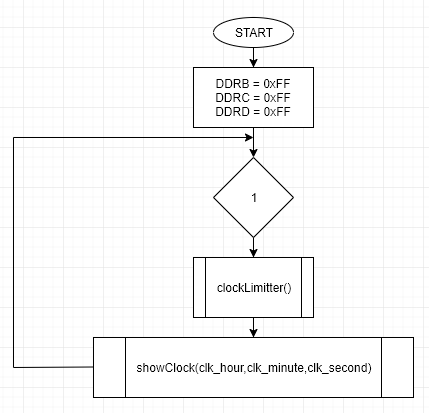
Hardware yang digunakan adalah :

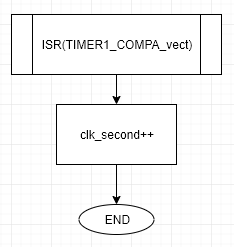
1. Arduino Nano dengan microprocessor ATMega 328P
2. 4 digit LED 7-segment x2
3. Transistor NPN 2N2222 x6
4. Resistor 1kΩ x6
5. PCB dot matriks
6. 9 V Battery cap
7. Kabel jumper
8. Female header 15x2
9. 9 V battery
10. 2 push button

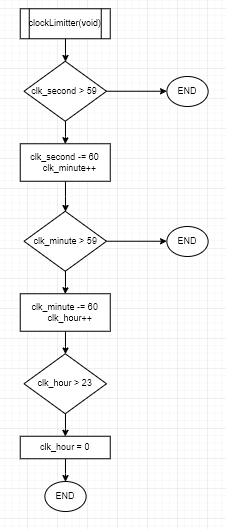
Pemilihan hardware dilakukan berdasarkan spesifikasi yang diminta. Diminta display 7 segment sebanyak 6 digit, maka digunakan 2 buah 4 digit 7 segment yang telah tersambung secara internal parallel untuk tiap modul. Keputusan untuk menggunakan 4 digit 7 segment x2 dikarenakan tidak terdapat 6 digit 7 segment dan 2 digit 7 segment yang tersedia di took terdekat dari ITB tidak tersambung secara internal parallel. Pemesanan secara online akan memakan biaya lebih banyak sehingga pilihan paling optimal yang terdapat pada saat pembuatan adalah menggunakan 4 digit 7 segment x2. Dikarenakan banyaknya led yang perlu dinyalakan dalam satu waktu, digunakan transistor agar nyala led tetap terang.

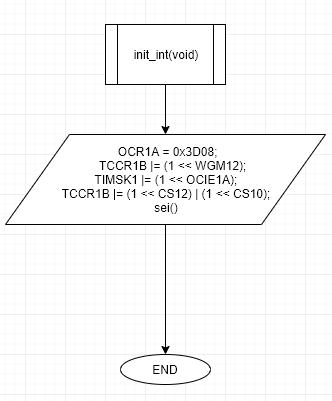
Penghubungan semua unit atau penyusunan rangkaian pada breadboard membutuhkan jumlah kabel yang cukup banyak sehingga kerapihan akan berkurang dan rawan terdapat pin yang tidak tersambung satu sama lain. Maka diputuskan untuk menggunakan pcb dot matriks karena waktu yang tersedia tidak begitu memungkinkan untuk mencetak pcb custom. Dimensi pcb dot matriks yang cukup kecil membuat opsi Arduino yang digunakan menjadi terbatas, dan diputuskan untuk menggunakan Arduino nano.

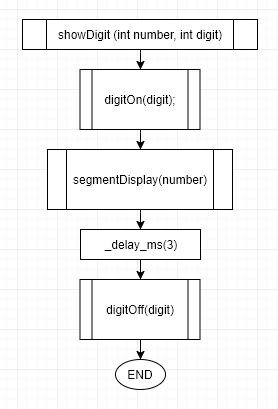
Skema software (flowchart)

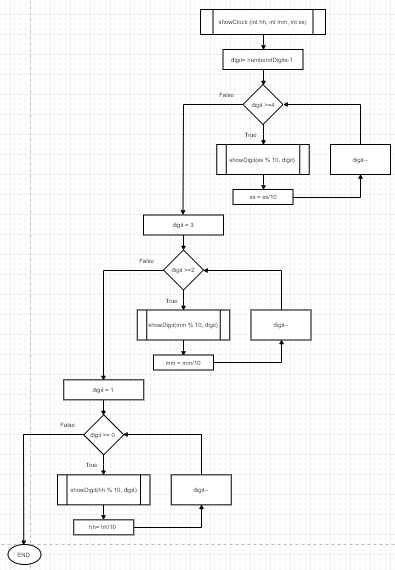


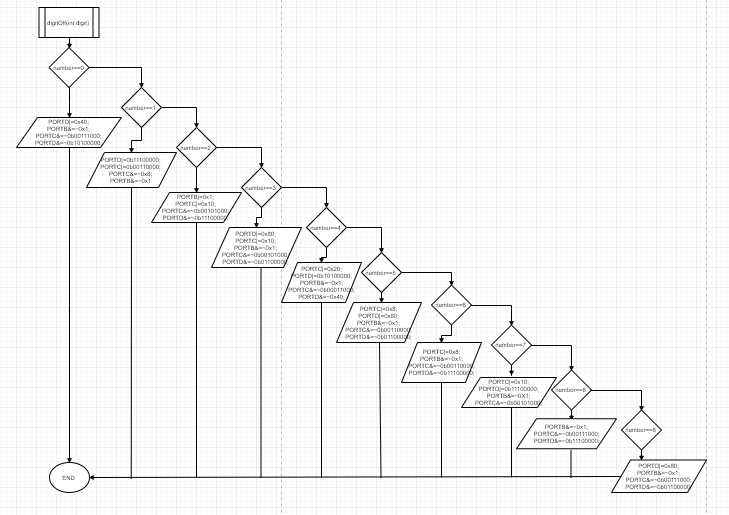


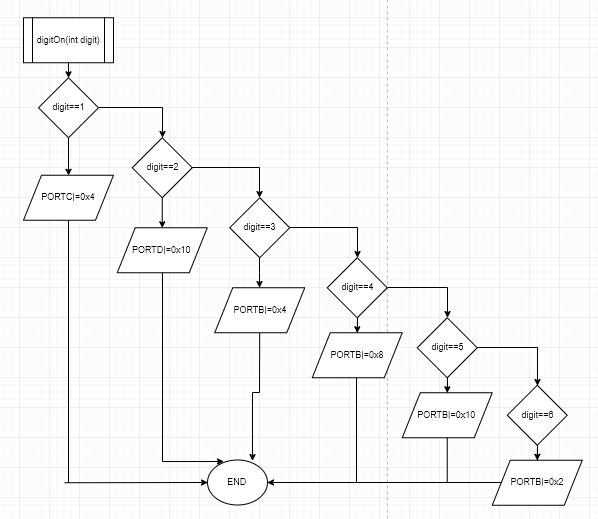


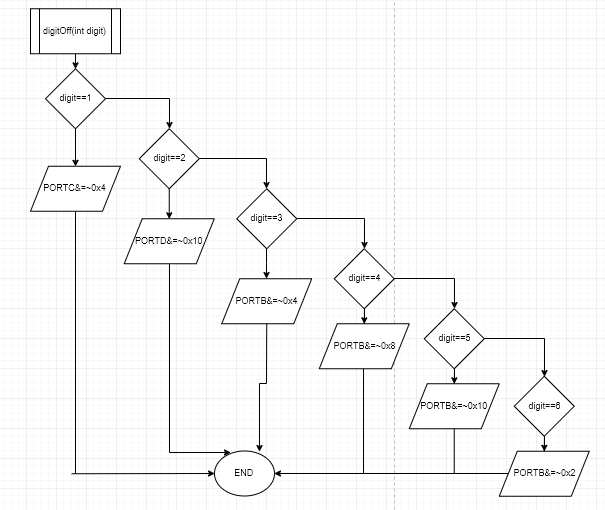








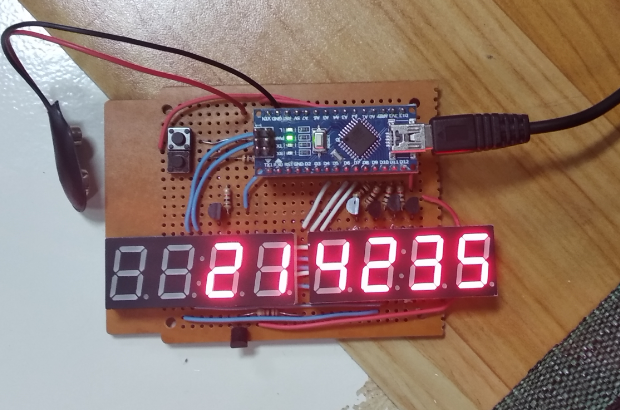




Penjelasan skema software

Program dimulai dengan inisialisasi timer dengan interrupt. Interupt yang digunakan dipicu dengan metode komparasi dan nilai timer yang digunakan sebesar 1 detik. Lalu diinisialisasi port B, C dan D sebagai output. Kemudian program memasuki looping tak hingga yang didalamnya menampilkan display jam digital. Display jam digital diatur dalam fungsi SHowClock() yaitu digit yang ditampilkan dinyalakan satu persatu sesuai waktu yang diharapkan. Agar display yang ditampilkan terlihat menyala bersamaan digunakan delay yang cukup kecil antara nyala digit 7 segment. Setiap detik, komparasi nilai OCR dan TCNT akan bernilai benar sehingga memicu pemanggilan interrupt yang berisikan penambahan nilai clk\_second sebesar 1. Di dalam looping tak hingga disisipkan pula fungsi clocklimitter() untuk membatasi nilai detik, menit dan jam.

Pengujian (hardware & software)



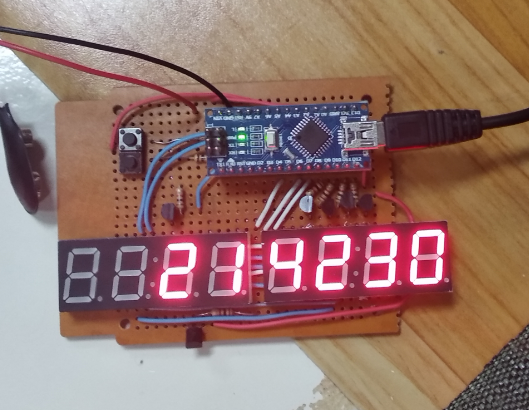
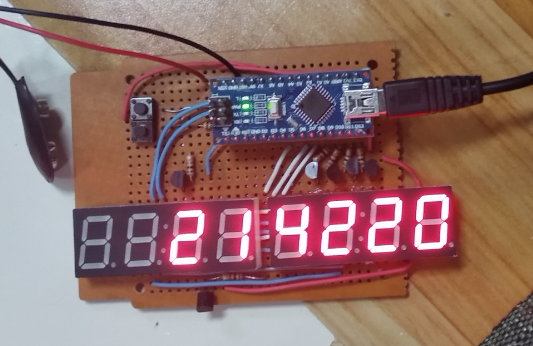


Foto alat



source code

|  |
| --- |
| */\**  *\* digital\_7\_segment\_clock.c*  *\**  *\* Initialized: 25/02/2019 16:35:38*  *\* Last edited: 19:19 Thursday,07 Maret 2019*  *\* Author : Muhammad Rafly Arya Putra (13216064)*  *Muhammad Irfaan S. (13216047)*  *Gusti Triandi Winata (13216091)*  *\* development repository : https://github.com/sanggusti/avr\_digital\_clock*  *\*/*  #define **CLOCK\_DISPLAY** 0  #define **HOUR\_SET** 1  #define **MINUTE\_SET** 2  #define **SECOND\_SET** 3  #define **MAX\_STATE** 4  #define **F\_CPU** 16000000L  #include <avr/io.h>  #include <util/delay.h>  #include <avr/interrupt.h>  volatile int state = CLOCK\_DISPLAY;  volatile int clk\_hour = 23;  volatile int clk\_minute = 59;  volatile int clk\_second = 30;  void **Init\_Ext\_Int**(void);  void **init\_int**(void);  void **clockLimitter**(void);  *//void showClock (int hh, int mm, int ss);*  void **showHour** (int hh);  void **showMinute** (int mm);  void **showSecond** (int ss);  void **showDigit** (int number, int digit);  void **digitOn**(int digit);  void **digitOff**(int digit);  void **segmentDisplay**(int number);  **ISR**(TIMER1\_COMPA\_vect)  {      if (state==CLOCK\_DISPLAY)          clk\_second++;  }  **ISR** (INT1\_vect)  {  *//function to change hour, minute, or second*      if (state==HOUR\_SET) {          clk\_hour++;          if (clk\_hour>23)              clk\_hour=0;      }      else if (state==MINUTE\_SET) {          clk\_minute++;          if (clk\_minute>59)              clk\_minute=0;      }      else if (state==SECOND\_SET) {          clk\_second++;          if (clk\_second>59)              clk\_second=0;      }  }  **ISR** (INT0\_vect)  {  *//function to change the state of the clock*      state++;      if (state>=MAX\_STATE)          state=CLOCK\_DISPLAY;  }  int **main**(void)  {  */\* inisialisasi output pin \*/*      DDRB = 0xFF;      DDRC = 0xFF;      DDRD = 0xFF;  */\* inisialisasi interrupt \*/*  **Init\_Ext\_Int**();  **init\_int**();      while (1) {          if (state==CLOCK\_DISPLAY) {  **clockLimitter**();  **showHour**(clk\_hour);  **showMinute**(clk\_minute);  **showSecond**(clk\_second);          }          else if (state==HOUR\_SET)  **showHour**(clk\_hour);          else if (state==MINUTE\_SET)  **showMinute**(clk\_minute);          else if (state==SECOND\_SET)  **showSecond**(clk\_second);      }        return 0;  }  void **init\_int**(void)  {  OCR1A = 0x3D08;  TCCR1B |= (1 << WGM12); *//Mode 4, CTC on OCR1A*  TIMSK1 |= (1 << OCIE1A); *//Mengatur interrupt untuk terpanggil ketika compare match*  TCCR1B |= (1 << CS12) | (1 << CS10); *//Mengatur prescaler menjadi 1024 dan memulai timer*  **sei**(); *// Mengaktifkan interrupt*  }  void **Init\_Ext\_Int**(void)  {      DDRD &= ~(1 << DDD2); *// Clear the PD2 pin*      DDRD &= ~(1 << DDD3); *// Clear the PD3 pin*  *// PD2 (PCINT0 pin) and PD3 (PCINT1) are now an input*      PORTD |= (1 << PORTD2); *// turn On the Pull-up*      PORTD |= (1 << PORTD3); *// turn On the Pull-up*  *// PD2 and PD3 are now an input with pull-up enabled*      EICRA |= (1 << ISC01); *// set INT0 to trigger on falling edge*      EICRA |= (1 << ISC11); *// set INT1 to trigger on falling edge*      EIMSK |= (1 << INT0); *// Turns on INT0*      EIMSK |= (1 << INT1); *// Turns on INT1*  }  void **clockLimitter**(void) {      if (clk\_second > 59) {          clk\_second -= 60;          clk\_minute++;          if (clk\_minute > 59) {              clk\_minute -= 60;              clk\_hour++;              if (clk\_hour > 23) {                  clk\_hour = 0;              }          }      }  }  void **showHour** (int hh) {      for (int digit= 2; digit >=1; digit--)      {  **showDigit**(hh % 10, digit);          hh= hh/10;      }  }  void **showMinute** (int mm) {      for (int digit= 4; digit >=3; digit--)      {  **showDigit**(mm % 10, digit);          mm= mm/10;      }  }  void **showSecond** (int ss) {      for (int digit= 6; digit >=5; digit--)      {  **showDigit**(ss % 10, digit);          ss= ss/10;      }  }  */\*void showClock (int \*hh, int \*mm, int \*ss) {*  *for (int digit= 6; digit >=5; digit--)*  *{*  *showDigit(ss % 10, digit);*  *ss= ss/10;*  *}*  *for (int digit= 4; digit >=3; digit--)*  *{*  *showDigit(mm % 10, digit);*  *mm= mm/10;*  *}*  *for (int digit= 2; digit >=1; digit--)*  *{*  *showDigit(hh % 10, digit);*  *hh= hh/10;*  *}*  *}\*/*  void **showDigit** (int number, int digit) {  **digitOn**(digit);  **segmentDisplay**(number);  **\_delay\_ms**(3);  **digitOff**(digit);  }  void **digitOn**(int digit) {      if (digit==1)          PORTC|=0x4;      else if (digit==2)          PORTD|=0x10;      else if (digit==3)          PORTB|=0x4;      else if (digit==4)          PORTB|=0x8;      else if (digit==5)          PORTB|=0x10;      else if (digit==6)          PORTB|=0x2;  }  void **digitOff**(int digit) {      if (digit==1)          PORTC&=~0x4;      else if (digit==2)          PORTD&=~0x10;      else if (digit==3)          PORTB&=~0x4;      else if (digit==4)          PORTB&=~0x8;      else if (digit==5)          PORTB&=~0x10;      else if (digit==6)          PORTB&=~0x2;  }  void **segmentDisplay**(int number) {      if (number==0) {          PORTD|=0x40;          PORTB&=~0x1;          PORTC&=~0b00111000;          PORTD&=~0b10100000;      }      else if (number==1) {          PORTD|=0b11100000;          PORTC|=0b00110000;          PORTC&=~0x8;          PORTB&=~0x1;      }      else if (number==2) {          PORTB|=0x1;          PORTC|=0x10;          PORTC&=~0b00101000;          PORTD&=~0b11100000;      }      else if (number==3) {          PORTD|=0x80;          PORTC|=0x10;          PORTB&=~0x1;          PORTC&=~0b00101000;          PORTD&=~0b01100000;      }      else if (number==4) {          PORTC|=0x20;          PORTD|=0b10100000;          PORTB&=~0x1;          PORTC&=~0b00011000;          PORTD&=~0x40;      }      else if (number==5) {          PORTC|=0x8;          PORTD|=0x80;          PORTB&=~0x1;          PORTC&=~0b00110000;          PORTD&=~0b01100000;      }      else if (number==6) {          PORTC|=0x8;          PORTB&=~0x1;          PORTC&=~0b00110000;          PORTD&=~0b11100000;      }      else if (number==7) {          PORTC|=0x10;          PORTD|=0b11100000;          PORTB&=~0X1;          PORTC&=~0b00101000;      }      else if (number==8) {          PORTB&=~0x1;          PORTC&=~0b00111000;          PORTD&=~0b11100000;      }      else if (number==9) {          PORTD|=0x80;          PORTB&=~0x1;          PORTC&=~0b00111000;          PORTD&=~0b01100000;      }  } |