

LAPORAN PROJECT
SISTEM MIKROPROSESOR
TWI PADA ARDUINO ATMEGA328



Disusun oleh :

Norman Mursalin

(04161050)

Yusril

(04161080)

INSTITUT TEKNOLOGI KALIMANTAN

DAFTAR ISI

DAFTAR ISI	i
BAB I LANDASAN TEORI	
1.1 Arduino	1
1.2 ATmega 328	1
1.3 Saluran I2C	2
1.4 1-Wire	3
BAB II PEMBAHASAN	
2.1 Hardware	4
2.2 Mode Pengelamatan (input/output)	5
2.3 Software	6
BAB III KESIMPULAN	7
DAFTAR PUSTAKA	8

BAB I

LANDASAN TEORI

1.1 Arduino

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino.

1.2 ATmega 328

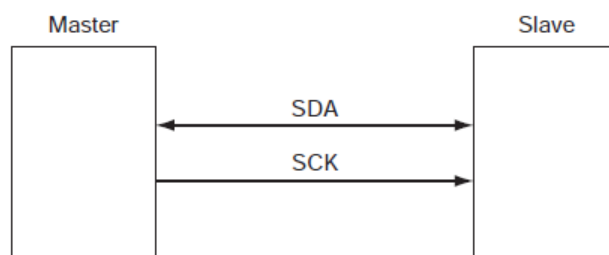
ATMega328 adalah mikrokontroler keluaran dari Atmel yang mempunyai arsitektur RISC (Reduce Instruction Set Computer) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (Completed Instruction Set Computer). ATMega 328 memiliki beberapa fitur antara lain : 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock. 32 x 8-bit register serbaguna. Kecepatan mencapai 16 MIPS dengan clock 16 MHz. 32 KB Flash memory dan pada Arduino memiliki bootloader yang menggunakan 2 KB dari flash memori sebagai bootloader. Memiliki EEPROM (Electrically Erasable Programmable Read Only Memory) sebesar 1 KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan. Memiliki SRAM (Static Random Access Memory) sebesar 2 KB. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (Pulse Width Modulation) output.

1.3 Saluran I2C

Seperti yang kita tahu, bahwa komunikasi yang paling banyak digunakan dalam system elektronika adalah jenis komunikasi serial. Alasan utamanya adalah komunikasi serial lebih sedikit memerlukan hardware sehingga menjadi lebih efisien. Awalnya, komunikasi I2C dikembangkan oleh Philips sekitartahun 1980-an dengan kecepatan transfer data yang masih kecil. Namun saat ini, kecepatan transfer data dari I2C sudah mencapai angka 4Mbps. Komunikasi serial I2C hanya memerlukan 2 kabel/saluran, sehingga sering juga disebut sebagai TWI/Two Wire Interface. Jika anda menggunakan metode komunikasi dengan I2C, anda tidak perlu repot untuk mengatur baudrate baik pada bagian pengirim data maupun pada bagian penerima data.

Komunikasi I2C (*Inter-Integrated Circuit*) merupakan koneksi dibuat untuk menyediakan komunikasi antara perangkat-perangkat terintegrasi, seperti sensor, RTC, dan juga EEPROM. Komunikasi I2C bersifat *synchronous* namun berbeda dengan SPI karena I2C menggunakan protocol dan hanya menggunakan dua kabel untuk komunikasi, yaitu *Synchronous clock* (SCL) dan *Synchronous data* (SDA). Secara berurutan data dikirim dari master ke slave kemudian (setelah komunikasi master ke slave selesai) dari *slave* ke *master*. Perangkat I2C menggunakan 2 buah pin *open-drain* duaarah dengan memberikan *pull-up* resistor untuk setiap garis *bus* sehingga berlaku seperti AND menggunakan kabel.

AVR dapat menggunakan 120 jenis perangkat untuk berbagi pada *bus* I2C yang masing-masing disebut sebagai *node*. Setiap *node* beroperasi sebagai *master* atau *slave*. Master merupakan perangkat yang menghasilkan *clock* untuk sistem, menginisiasi, dan juga memutuskan sebuah transmisi. *Slave* merupakan *node* yang menerima *clock* dan dialamatkan oleh *master*. Baik *master* dan *slave* dapat menerima dan mentransmisikan data.



Gambar 1 Aliran data I2C

I2C merupakan protocol komunikasi serial dimana setiap bit data ditransfer pada jalur SDA yang disinkronisasikan dengan pulsa *clock* pada jalur SCL. Jalur data tidak dapat berubah ketika jalur *clock* berada dalam kondisi *high*.

Dalam I2C, setiap alamat atau data yang ditransmisikan harus dibentuk dalam sebuah paket dengan panjang 9 bit dimana 8 bit pertama disimpan dalam jalur SDA oleh *transmitter*, dan bit ke-9 merupakan *acknowledge* (atau *not acknowledge*) oleh *receiver*. I2C juga diistilahkan sebagai *Two-wire Serial Interface* (TWI), bergantung dari istilah yang digunakan oleh pabrik yang membuat perangkat. Salah satu perangkat yang digunakan dengan komunikasi I2C adalah *real-time clock* (RTC). Perangkat ini menyediakan komponen jam, menit, dan detik, serta tahun, bulan, dan hari.

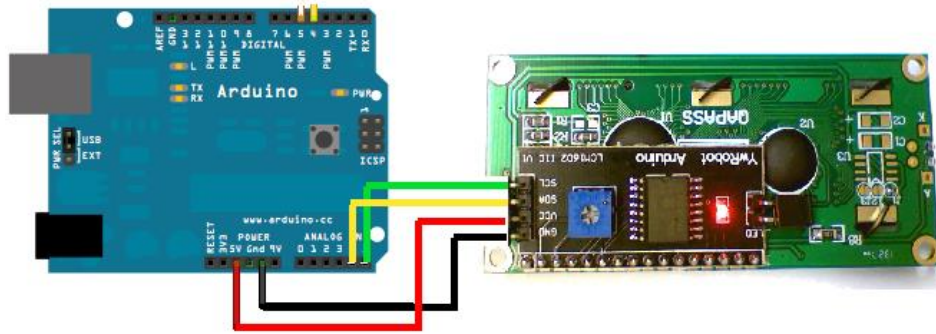
1.4 1-wire

Komunikasi *1-wire* merupakan perangkat unik yang hanya membutuhkan sebuah kabel (dan tambahan kabel *ground*) untuk melakukan komunikasi dengan sebuah perangkat. Komunikasi yang dilakukan merupakan komunikasi *asynchronous* yang mengikuti skema *master/slave*. Satu buah *master* dapat digunakan untuk terhubung dengan beberapa *slave*.

BAB II

PEMBAHASAN

2.1 Hardware



Gambar 2. Jalur Komunikasi I2C

Komunikasi I2C mikrokontroler ataupun I2C pada arduino secara garis besar memiliki prinsip kerja yang sama. I2C yang merupakan singkatan dari inter integrated circuit adalah komunikasi serial dan sinkron bus protokol yang memungkinkan kita untuk menghubungkan sejumlah device (*slave*) dengan device utama (*master*) dalam satu mode komunikasi.

Seperti telah dijelaskan di atas, sistem komunikasi I2C terdiri dari satu master dan slave (bisa lebih dari satu). Master berfungsi untuk membangkitkan dan mengirim *clock pulse* serta menentukan kapan dan dengan *slave device* mana berkomunikasi.

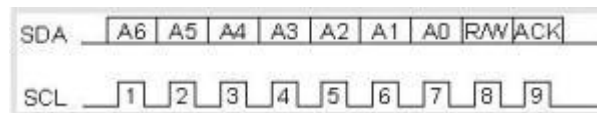
Berdasarkan diagram komunikasi I2c diatas, transfer data dimulai dengan *start condition* yaitu kondisi dimana SCL bernilai high dan SDA mengalami perubahan nilai dari high ke low. Lalu master memilih device (*slave*) tujuan berkomunikasi dengan mengidentifikasi 8 bit adress *slave*.

Lalu setelah itu, master mentransfer data “1 (high)” jika ingin membaca data dari *slave* dan mentransfer data “0 (low)” jika ingin memberi perintah pada device *slave*. Setelah itu, master dapat berkomunikasi dengan memberi perintah atau mengambil data dari device (*slave*), lalu diakhiri dengan *stop condition* dimana kondisi sinyalnya berlawanan dengan kondisi *start condition*.

2.2 Mode Pengalamatan (input/output)

Hal yang penting lagi dalam proses komunikasi I2C adalah mode pengalamatan dari master dan slave, terlebih jika ada beberapa master/beberapa slave dalam 1 bus tersebut. Mode pengalamatan memungkinkan kita untuk mendapatkan data yang benar sesuai dengan data yang tersimpan di dalam komponen tersebut.

Mode pengalamatan terdiri 7 bit + 1 bit R/W + 1bit ACK. Jika 1 device alamat terdiri 7 bit data, maka logikanya dalam 1 bus bisa dipasang sampai dengan 127 device. Pengiriman data pertama dalam I2C adalah MSB (Most Significant Bit) dan terakhir adalah LSB (Least Significant Bit) kemudian dilanjutkan dengan 1 bit ACK.



Gambar 3. Model pengalamatan I2C

A6 adalah MSB, R/W adalah LSB. Bit R/W bernilai 1 jika master akan membaca data, dan bernilai 0 jika ingin menulis data.

Ada kalanya dalam suatu system kita memerlukan lebih dari 1 Microcontroller yang saling bertukar data/berkomunikasi. Prinsip master dan slave tetap berlaku meski proses komunikasi dilakukan oleh microcontroller. Yakni clock berasal dari master sebagai sinkronisasi data. Kendala utamanya adalah, slave tidak bisa langsung mengirim data, karena slave juga memerlukan waktu untuk ke dalam interrupt service routine, menyimpan data ke dalam variable, dan menyesuaikan alamat data yang diminta oleh master. Beberapa hal tersebut memerlukan waktu, sehingga jika master tetap saja mengirim clock maka data yang diterima dari slave akan error/respon yang diberikan salah.

Untuk mengatasi hal ini, diperlukan saling pengertian antara microcontroller master dengan microcontroller slave. Caranya, slave bisa menahan SCL pada posisi Low saat pertama menerima perintah read dari master, kemudian slave menyiapkan data yang diminta, setelah siap maka slave melepas SCL untuk kembali menjadi High. Sedangkan pada posisi master, perintah clock

pada SCL untuk membaca slave juga diikuti dengan recheck SCL apakah pada tetap pada posisi Low ataukah sudah kembali ke High, jika sudah High artinya data pada slave siap dikirim dan master bisa melanjutkan pengambilan datanya dengan mengirim clock selanjutnya.

2.3 Software

Berikut merupakan contoh program pada I2C Arduino yang menggunakan LCD sebagai slave dari program



Gambar 4. Arduino dan LCD

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup() {
  lcd.begin();}

void loop(){
  lcd.setCursor(0,0);
  lcd.print("TEST LCD i2C");
  lcd.setCursor(0,1);
  lcd.print("KelasRobot.com");
}
```


BAB III

KESIMPULAN

Kita dapat menarik kesimpulan bahwa, komunikasi TWI/I2C adalah jenis komunikasi yang hanya menggunakan dua serial, yaitu serial data dan serial clock. Sistem ini dapat menggunakan satu master untuk beberapa slave, dimana master yang mengatur kapan slave akan membaca atau mengirim data. Namun, slave dapat menahan clock dari master jika belum selesai melaksanakan perintah sebelumnya.

DAFTAR PUSTAKA

8-bit Atmel Microcontroller with 128Kbytes In-System Programmable Flash:
ATMEGA128

Atmel, AVR318: Dallas 1-Wire Master on tinyAVR and megaAVR

Barnett, Cull, Cox. 2007. *Embedded C Programming and the Atmel AVR 2nd ed.*
Nelson Education, Ltd.: Canada

Maxim Integrated, *Choosing the Right 1-Wire® Master for Embedded Applications*

Mazidi, Muhammad Ali. 2011. *The Microcontroller and Embedded System: Using Assembly and C.* Pearson Education, inc: New Jersey