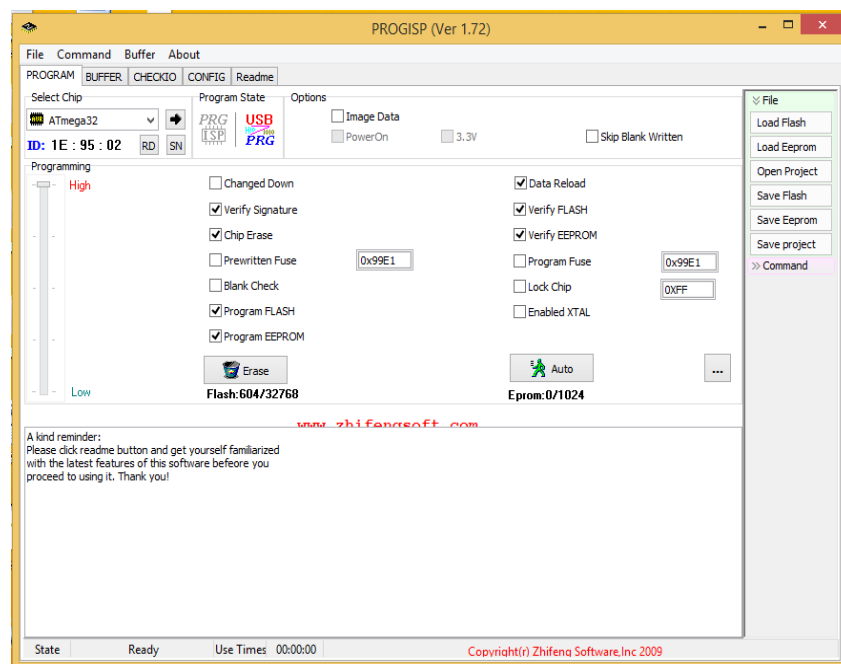


BAB V

ANALISA DAN PENGUJIAN SISTEM

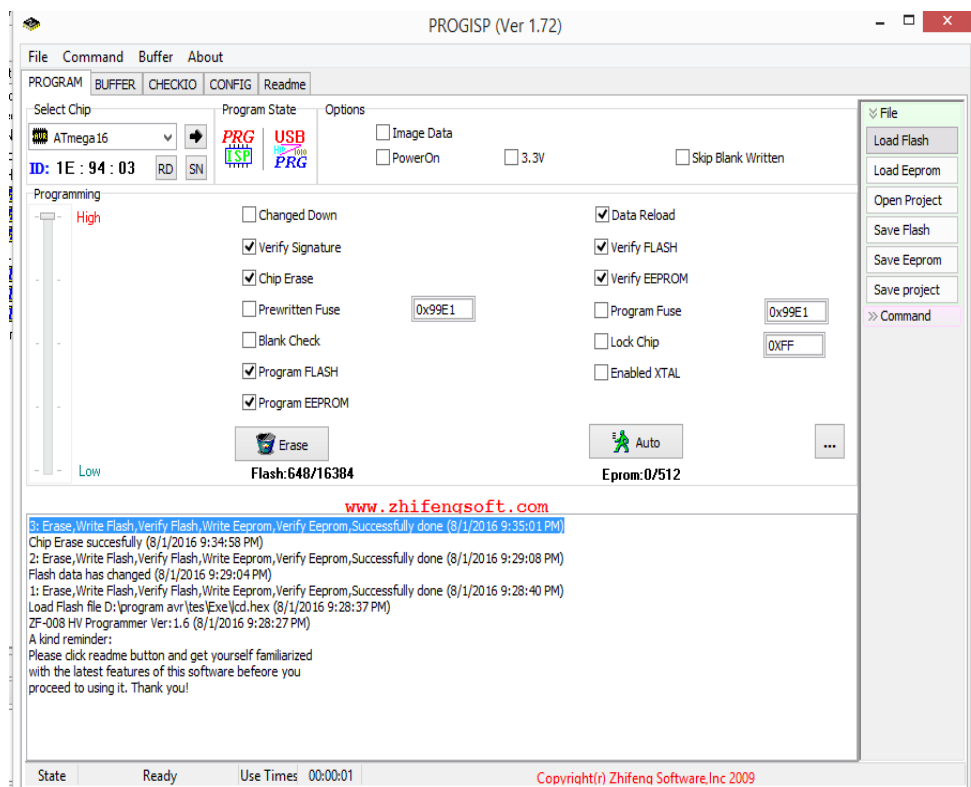
5.1 PENGUJIAN RANGKAIAN MIKROKONTROLER ATMEGA32

Pengujian pada rangkaian mikrokontroler ini menggunakan *software* PROGISP (Ver 1.72). Pengujian rangkaian ini bertujuan untuk mengetahui apakah mikrokontroler telah aktif atau tidak aktif. Mikrokontroler harus dapat diprogram langsung pada papan rangkaian dan rangkaian mikrokontroler harus dapat dikenali oleh program *downloader*. Pada pengujian ini, yang pertama harus dilakukan adalah menghubungkan rangkaian mikrokontroler ke *PC* dengan menggunakan *downloader*, kemudian buka *software* PROGISP dan pilih *chip select* ATMega16 seperti yang ditampilkan pada gambar 5.1.



Gambar 5.1 Tampilan *software* PROGISP (Ver 1.72)

Langkah selanjutnya adalah dengan mengklik *Read Signature* pada bagian *toolbar* kanan maka *textbox* akan menampilkan *Read ID Successfully*. Jika sudah tampil pada *textbox* kata *Read ID Successfully* seperti tampilan pada gambar 5.2 maka mikrokontroler telah aktif dan siap untuk digunakan.



Gambar 5.2 Mikrokontroler Telah Terdeteksi

5.2 PENGUJIAN MOTOR DC

Pada tahap ini peneliti melakukan pengujian pada motor DC untuk melihat pergerakan dari motor DC yang telah terhubung ke IC I293D, untuk lebih jelas dapat dilihat pada tabel pengujian 5.1 :

Tabel 5.1 Pengujian Motor DC

No	Arus Motor 1	Arus Motor 2	Motor 1		Motor 2		KET.
			PORT A.0	PORT A.1	PORT A.2	PORT A.3	
1	12 V	12 V	1	0	1	0	Bergerak Maju
2	12 V	12 V	0	1	0	1	Bergerak Mundur
3	0 V	0 V	0	0	0	0	Berhenti
4	12 V	0 V	1	0	0	0	Belok Kanan
5	0 V	12 V	0	0	1	0	Belok Kiri

Dari tabel diatas dapat dilihat jika motor 1 dan motor 2 sama – sama diberi logika “1” dan “0” maka motor dc akan bergerak maju dan jika motor 1 dan 2 diberi logika “0” dan “1” maka motor dc pada robot akan bergerak mundur. Apabila salah motor dc diberi logika “0” dan “0” maka motor dc akan diam atau tidak bergerak.

5.3 PENGUJIAN RANGKAIAN LCD

Bagian ini hanya terdiri dari sebuah *LCD* dot matriks 16x2 karakter. Pengujian ini dilakukan untuk menunjukkan tampilan hasil pengukuran sensor MQ-138 dan tampilan dari beberapa keterangan lainnya. *Display* karakter pada *LCD* diatur oleh pin *EN*, *RS* dan *RW*. Jalur *EN* dinamakan *Enable*. Jalur ini digunakan untuk memberitahu *LCD* bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke *LCD*, maka melalui program *EN* harus dibuat logika *low* ‘0’ dan set *high* ‘1’ pada dua jalur kontrol yang lain *RS* dan *RW*. Jalur *RW* adalah jalur kontrol *Read/Write*. Ketika *RW* berlogika *low* ‘0’, maka informasi pada *bus* data akan dituliskan pada layar *LCD*. Ketika *RW* berlogika *high* ‘1’,

maka program akan melakukan pembacaan memori dari *LCD*. Sedangkan pada aplikasi umum pin *RW* selalu diberi logika *low* '0'.

Berdasarkan keterangan di atas maka kita sudah dapat membuat program untuk menampilkan karakter pada *display LCD*. Adapun program yang diisikan ke mikrokontroler untuk menampilkan karakter pada *display LCD* adalah sebagai berikut:

```
#include <mega32.h>
#include <alcd.h>
lcd_init(16);
while (1)
{
    // Place your code here
    lcd_gotoxy(5,0);
    lcd_putsf("TES LCD");
    lcd_gotoxy(1,1);
    lcd_putsf("STIKOM-DB-JAMBI");
}
```

Program di atas akan menampilkan kata "TES LCD" di baris pertama pada *display LCD* 16x2 dan "STIKOM-DB-JAMBI" pada baris kedua *LCD* 16x2.

Untuk tampilan pengujian *LCD* dapat dilihat pada gambar 5.3.



GAMBAR 5.3 *Display Pengujian LCD 16X2*

5.4 PENGUJIAN SENSOR PHOTODIODA

Pengujian sistem input pada sensor *photodiode* dalam membaca garis lintasan dengan menggunakan 8 buah pasang *LED* dan *Photodiode* yang di hubungkan ke mikrokontroler.

Pengujian sensor *Photodiode* meyangkut pada bagian sensor yang berfungsi untuk membaca lintasan dengan modul program. Pengujian ini dilakukan untuk mengetahui bahwa sensor dapat membaca lintasan dengan baik. Untuk lintasan yang berwarna hitam sensor akan mengirim nilai logika '1' kepada mikrokontroler dan untuk lintasan berwarna putih sensor akan mengirim nilai logika '0' kepada mikrokontroler. Jika dilakukan pengecekan terhadap sensor, maka sensor yang terkena garis hitam akan bernilai '1' dan yang terkena bagian putih akan bernilai '0', ini menandakan pengujian terhadap sensor sudah berjalan dengan baik dapat dilihat pada table 5.2

Tabel 5.2 Pengujian Sensor

No	Output LCD	Keadaan Sensor							
		Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
1	00000000	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH
2	00000001	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	HITAM
3	00000011	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	HITAM	HITAM
4	00000111	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	HITAM	HITAM	HITAM
5	00001111	PUTIH	PUTIH	PUTIH	PUTIH	HITAM	HITAM	HITAM	HITAM
6	00011111	PUTIH	PUTIH	PUTIH	HITAM	HITAM	HITAM	HITAM	HITAM
7	00111111	PUTIH	PUTIH	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM
8	01111111	PUTIH	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM
9	11111111	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM
10	10000000	HITAM	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH
11	11000000	HITAM	HITAM	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH
12	11100000	HITAM	HITAM	HITAM	PUTIH	PUTIH	PUTIH	PUTIH	PUTIH
13	11110000	HITAM	HITAM	HITAM	HITAM	PUTIH	PUTIH	PUTIH	PUTIH
14	11111000	HITAM	HITAM	HITAM	HITAM	HITAM	PUTIH	PUTIH	PUTIH
15	11111100	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	PUTIH	PUTIH

16	11111110	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	PUTIH
17	11111111	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM	HITAM

5.5 PENGUJIAN KESELURUHAN

Tabel 5.3 Pengujian Keseluruhan

No	Pushbutton	Tujuan Meja	Waktu (s)		
			Pergi	kembali	keseluruhan
1	1	1	± 21 s	± 9 s	± 30 s
2	2	2	± 35 s	± 20 s	± 55 s
3	3	3	± 21 s	± 30 s	± 51 s
4	4	4	± 45 s	± 30 s	± 75 s

Pada table 5.3 dapat diketahui bahwa percobaan robot mengantarkan makanan menuju ke meja tujuan berhasil dengan waktu keberangkatan dan kembali ke *home base* yang bervariasi. Adanya waktu yang bervariasi pada saat berangkat dan kembali menunjukkan bahwa robot memerlukan waktu yang bervariasi sesuai dengan kecepatan putaran Motor DC dan sensor garis dalam membaca warna untuk robot berjalan sesuai dengan program dari mikrokontroler ATmega32.