Application Note

Maze Runner Robot

AN-CM-215

Abstract

This application shows how to interpret data from different sensors and how to design a robot to autonomously solve a maze with a GreenPAK™ IC.

This application note comes complete with design files which can be found in the References section.

Contents

Figures

Tables

1 Terms and Definitions

ASM	Asynchronous State Machine
IC	Integrated circuit
IEEE	Institute of electrical and electronics engineers
IR	Infrared

2 References

For related documents and software, please visit:

GreenPAK™ Programmable Mixed-Signal Products | Renesas

Download our free GreenPAK Designer software [1] to open the .gp files [2] and view the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Renesas Electronics provides a complete library of application notes [3] featuring design examples as well as explanations of features and blocks within the Renesas Electronics IC.

[1] GreenPAK Designer Software, Software Download and User Guide, Renesas Electronics
[2] AN-CM-215 Maze Runner Robot.gp, GreenPAK Design File, Renesas Electronics
[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage, Renesas Electronics
[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage, Renesas Electronics

3 Introduction

Maze-solving robots originate from the 1970s. Since then, the IEEE has been holding maze solving competitions called the Micro Mouse Contest. The aim of the contest is to design a robot that finds the midpoint of a maze as quickly as possible. The algorithms used to quickly solve the maze typically fall into three categories; random search, maze mapping, and right or left wall following methods.

The most functional of these methods is the wall following method. In this method, the robot follows the right or left side wall in the maze. If the exit point is connected to the outer walls of the maze, the robot will find the exit. This app note uses the right wall following method.

4 Hardware

This application uses:

- 2 Sharp analog distance sensors
- Tracker sensor
- Encoder
- Motors and motor driver
- GreenPAK™ SLG46531V
- Voltage regulator, robot chassis etc.

We will use the analog distance sensor to determine the distances to the right and front walls. The Sharp distance sensors are a popular choice for many projects that require accurate distance measurements. This IR sensor is more economical than sonar rangefinders, yet it provides much better performance than other IR alternatives.

There is a nonlinear, inverse relationship between the output voltage of the sensor and the measured distance. The plot showing the relationship between the sensor output and the measured distance is shown in Figure 1.

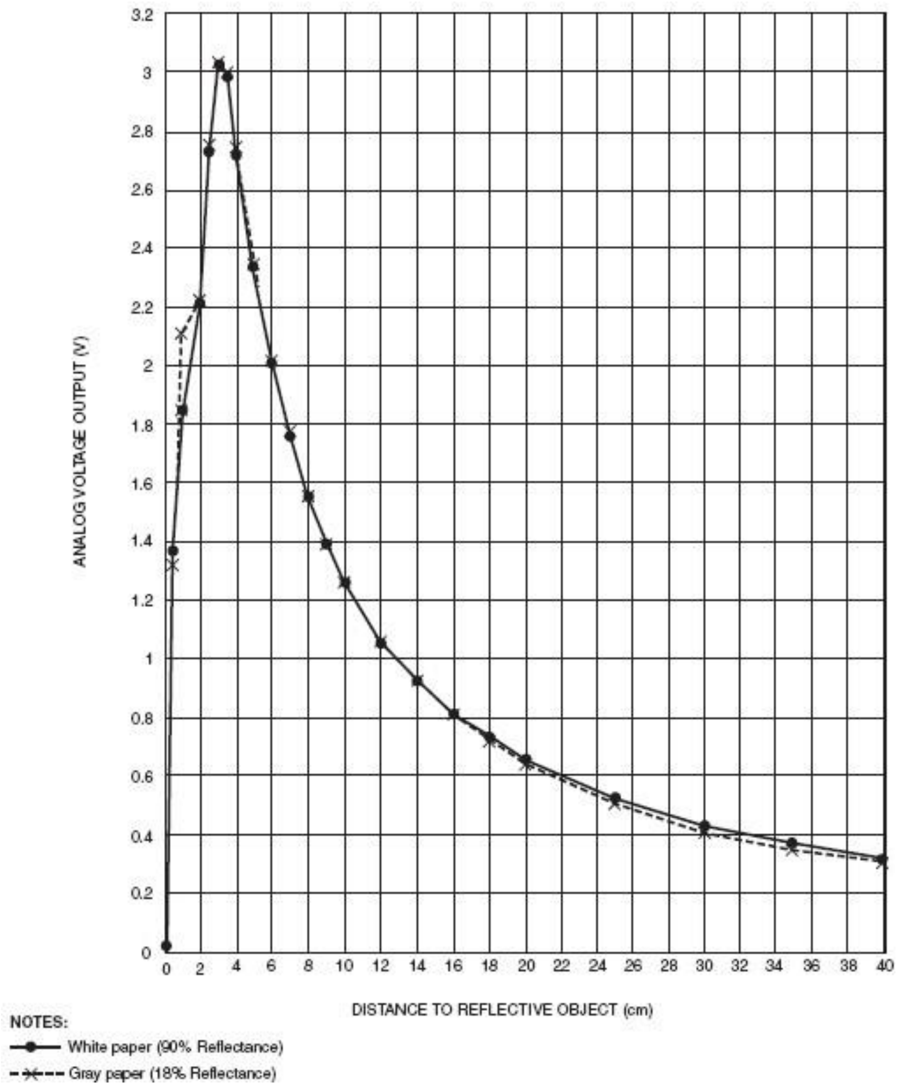


Figure 1: Distance-Voltage Graph

A white line against a black color ground is set as the target. We will use the tracker sensor to detect the white line. The tracker sensor has five analog outputs, and the outputted data is influenced by the distance and the color of the detected object. The detected points with higher infrared reflectance (white) will cause a higher output value, and the lower infrared reflectance (black) will cause a lower output value.

Figure 2: Tracker Sensor

We will use the pololu wheel encoder to calculate the distance the robot travels. This quadrature encoder board is designed to work with pololu micro metal gearmotors. It functions by holding two infrared reflectance sensors inside the hub of a Pololu 42 mm ×19 mm wheel and measuring the movement of the twelve teeth along the wheel's rim.

Figure 3: Pololu Wheel Encoder, Motor and Wheel

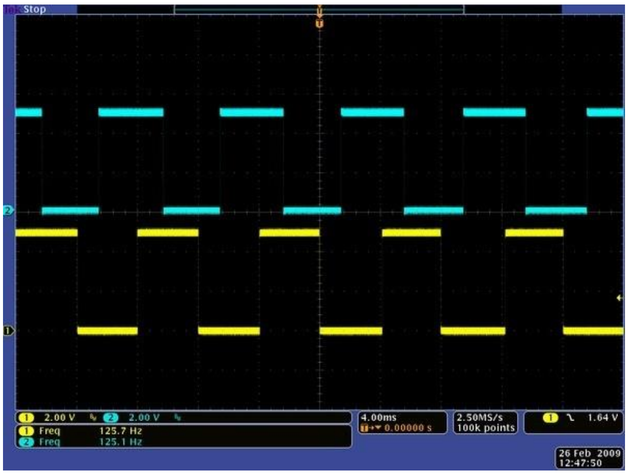


Figure 4: Encoder Outputs

A motor driver circuit board (L298N) is used to control the motors. The INx pins are used to direct the motors, and the ENx pins are used to set the speed of the motors.

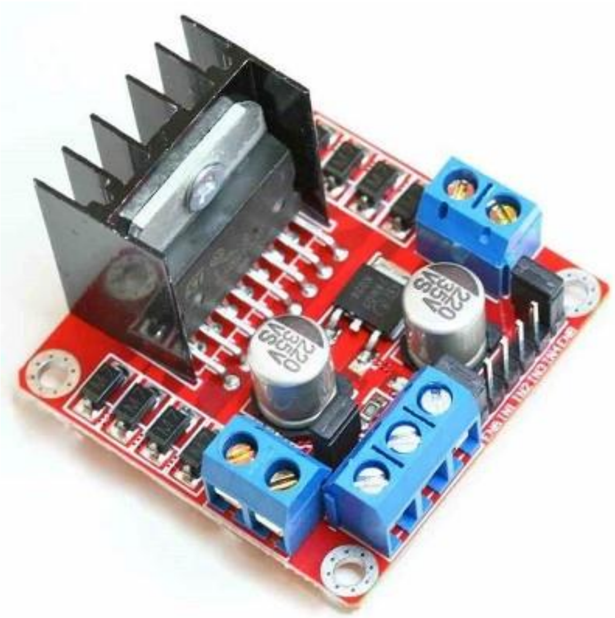


Figure 5: Motor Driver Circuit Board

Also, a voltage regulator is used to reduce the voltage from the battery down to 5 V.

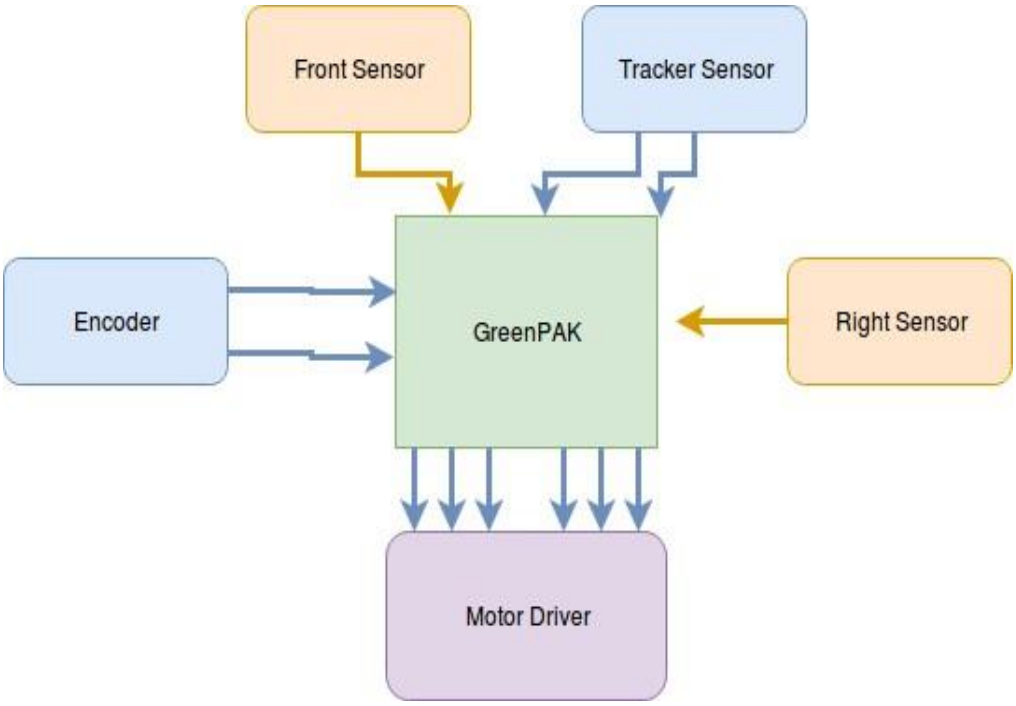


Figure 6: Inputs and Outputs of GreenPAK Chip

5 Algorithm Description

This app note incorporates the right wall following method. This is based upon organizing direction priority by preferring the rightmost possible direction. If the robot cannot detect the wall on the right, it turns to the right. If the robot detects the right wall and there is no wall in front, it goes forward. If there is a wall to the right of the robot and the front, it turns to the left.

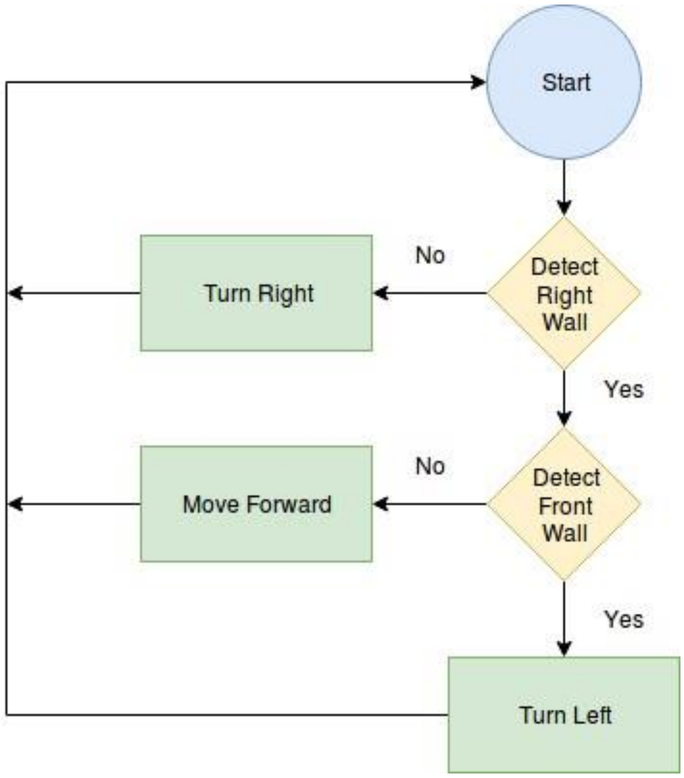


Figure 7: Simplified Algorithm

An important note is that there is no wall for reference after the robot has just turned to the right.

Therefore “turning right” is accomplished in three steps. Move forward, turn right, move forward.

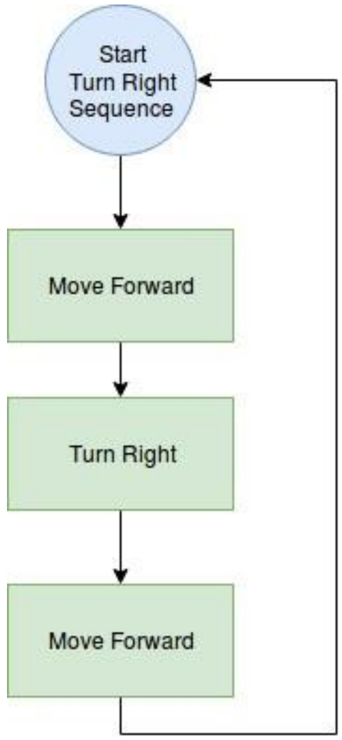


Figure 8: Turn Right Sequence

In addition, the robot must keep its distance from the wall when moving forward. This can be done by adjusting one motor to be faster or slower than the other.

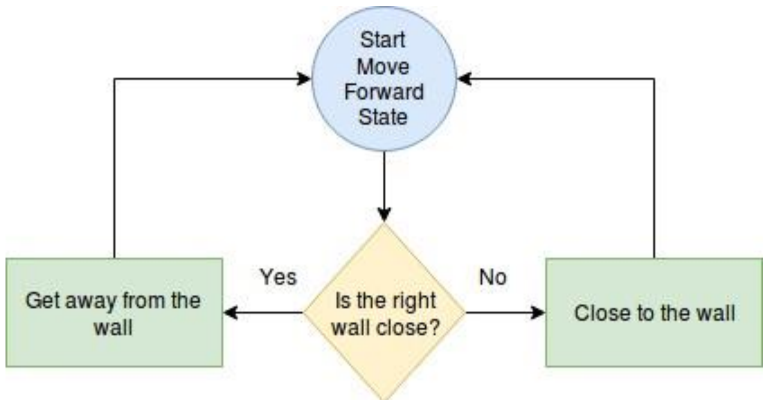


Figure 9: Collision Avoidance Algorithm

The final state of the flow chart is shown in Figure 10.

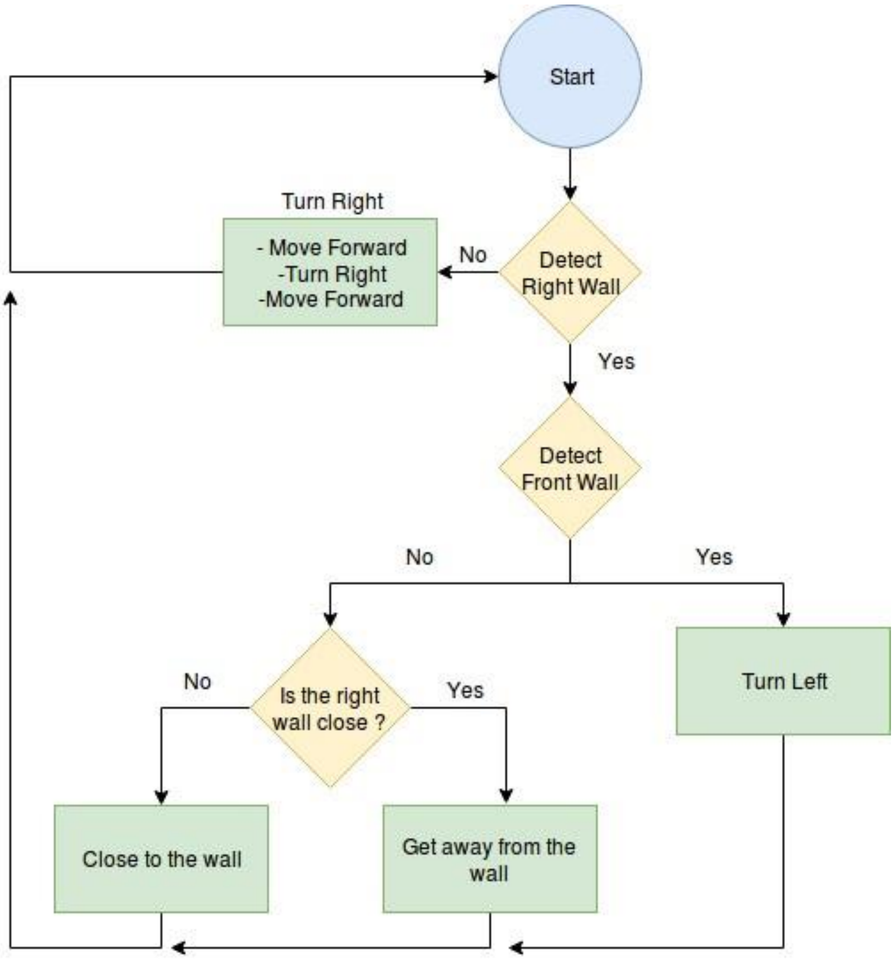


Figure 10: Final Algorithm GreenPAK

6 Design

The GreenPAK design consists of two parts. These are:

- Interpretation / processing of data from distance sensors ASM
- states and motor outputs

6.1 Interpretation / Processing of Data from Distance Sensors

It is important to interpret the data from the distance sensors. The robot's movements are deliberated according to the distance sensors outputs. Since the distance sensors are analog, we will use the ACMPs. The position of the robot relative to the wall is determined by comparing the voltages of the sensors with the predetermined threshold voltages.

We will use 3 ACMPs;

- To detect the front wall (ACMP2)
- To detect the right wall (ACMP0)
- To protect the distance of the right wall (ACMP1)

Since ACMP0 and ACMP1 depend on the same distance sensor, we used the same IN+ source for both comparators. Constant signal change can be prevented by giving ACMP1 25mv of hysteresis.

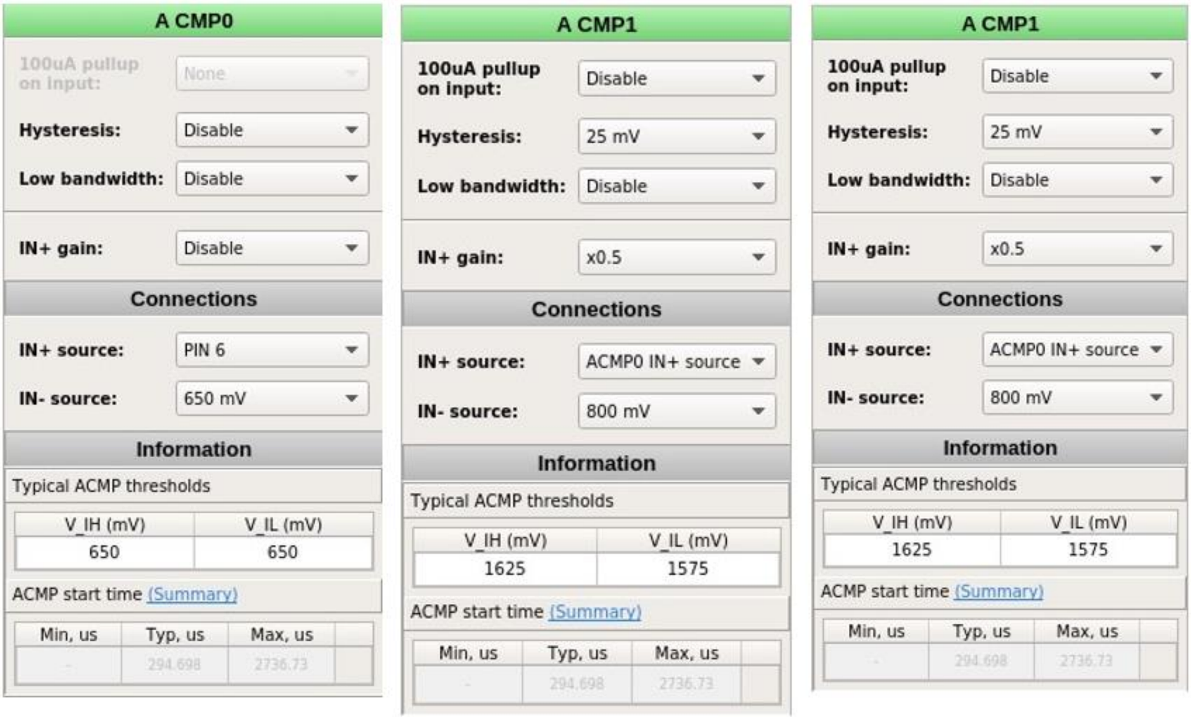


Figure 11: ACMP's Settings

We can determine the direction signals based on the ACMPs' outputs. The circuit shown in Figure 12 depicts the flow diagram outlined in Figure 7.

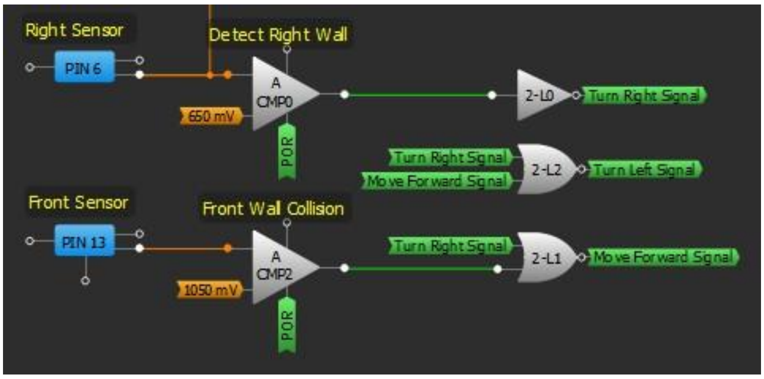


Figure 12: Wall Detection Circuit

In the same way, the circuit which indicates the position of the robot relative to the right wall is shown in Figure 13.

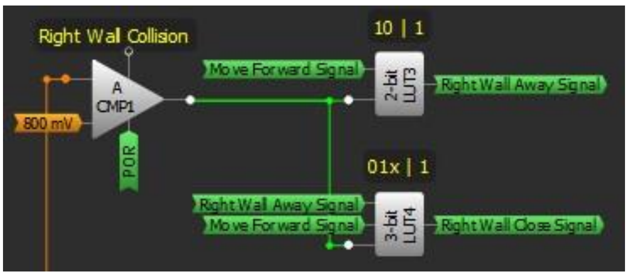


Figure 13: Wall Collision Circuit

6.2 ASM States and motor outputs

This application utilizes the Asynchronous State Machine, or ASM, to control the robot. There are 8 states in the ASM, and 8 outputs in each state. The Output RAM can be used to adjust these outputs. The states are listed below:

- Start
- Control
- Move away from the right wall
- Close to the right wall
- Turn Left
- Move Forward-1
- Turn Right
- Move Forward-2

These states determine the output to the motor driver and direct the robot. There are 3 outputs from the GreenPAK for each motor. Two determine the direction of the motor, and the other output determines the speed of the motor. The motor movement according to these outputs is shown in the following tables:

Table 1: Motor Movement According to Inputs

IN1	IN2	Definition
0	0	Motors remain off
0	1	Motor rotates clockwise
1	0	Motor rotates Counter-clockwise
1	1	Motors remain off

Table 2: PWM Table

PWM	Definition
0	High Speed
1	Low Speed

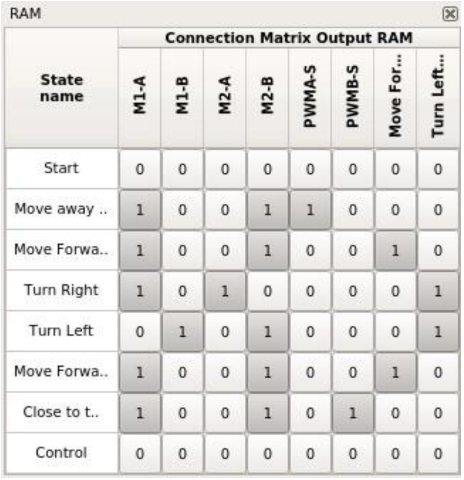
Table 3: Robot Movement According to Inputs

M1-A	M1-B	M2-A	M2-B	Movement
0	0	0	0	Stop
1	0	0	1	Move Forward
0	1	1	0	Move Backward
1	0	1	0	Turn Right
0	1	0	1	Turn Left

Table 4: Robot Movement with PWM

M1-A	M1-B	M2-A	M2-B	PWMA	PWMB	Movement
1	0	0	1	0	1	Close to the right wall
1	0	0	1	1	0	Move away from the right wall

The ASM Output RAM is derived from these tables. It is shown in Figure 14:



RAM

State name	M1-A	M1-B	M2-A	M2-B	PWMA-S	PWMB-S	Move For...	Turn Left...
Start	0	0	0	0	0	0	0	0
Move away ..	1	0	0	1	1	0	0	0
Move Forwa..	1	0	0	1	0	0	1	0
Turn Right	1	0	1	0	0	0	0	1
Turn Left	0	1	0	1	0	0	0	1
Move Forwa..	1	0	0	1	0	0	1	0
Close to t..	1	0	0	1	0	1	0	0
Control	0	0	0	0	0	0	0	0

Figure 14: Connection Matrix Output RAM

In addition to the motor drivers there are two more outputs. These outputs go to the corresponding delay blocks to allow the robot to travel a certain distance.

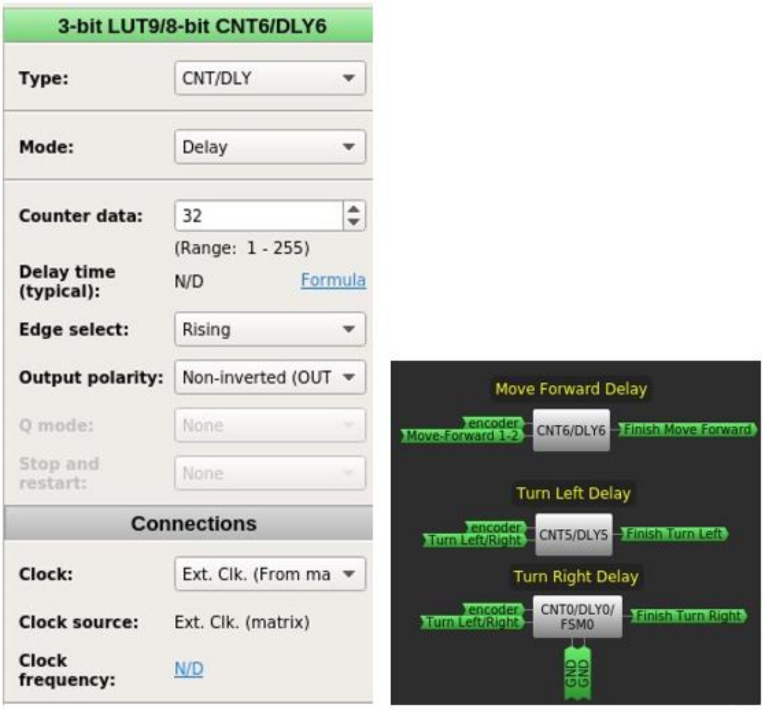


Figure 15: Move Forward Delay Settings and Circuits

The outputs of these delay blocks are also connected to ASM inputs.

Figure 16: ASM Inputs and Outputs

PWMs were used to adjust the speed of the motors. The ASM was used to determine what PWM the motor would run on. The PWMA-S and PWMB-S signals are set to the mux select bits.

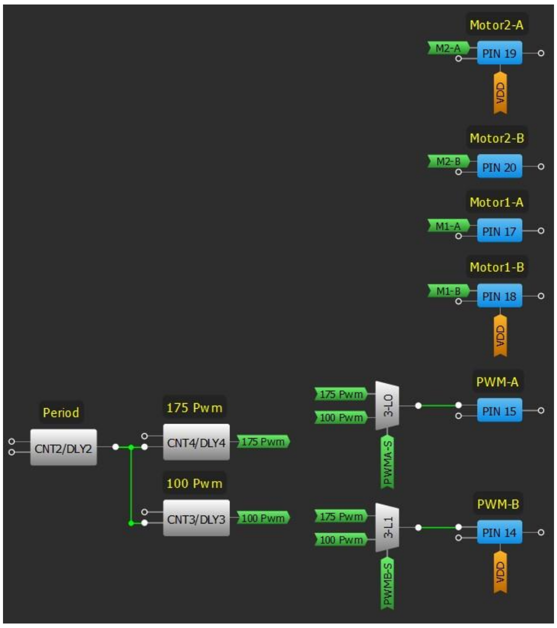


Figure 17: Motor Outputs

7 Conclusion

In this project, we used a GreenPAK SLG46531V to create a maze-solving robot. We interpreted data from multiple sensors, controlled the robot's state with the GreenPAK's ASM, and drove the motors with a motor driver. Generally, microprocessors are used in such projects, but a GreenPAK has a few advantages over an MCU: it is smaller, more affordable, and can process the sensor output faster than an MCU.

Revision History

Revision	Date	Description
1.0	16-Nov-2017	Initial Version

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.