

Chapter 2

Propeller Microcontroller

On successful completion of this course, students will be able to:

- Describe some of popular microcontrollers.
- Explain how to program the Propeller Microcontroller.
- Assembly a simple mobile robot using microcontroller.

Introduction

Microcontroller is the main controller for electronic devices today, including robots. Microcontroller well-known and readily available today are AVR, PIC, Arduino, Propeller, ATmega 8535, ATmega16, ATmega32 and Basic Stamp. Some other well-known brands eg 16F877 PIC and Basic Stamp 2. Parallax Propeller microcontroller from one of the latest generation 32-bit microcontroller that is capable of computing high-speed data. This microcontroller has many advantages especially can be used for image processing. Therefore, this microcontroller is used as the main control system of our robot.

Introduction of Propeller Chip

Do you like programming? With eight 32-bit processors in one chip, integrating peripheral devices is suddenly simplified with the Propeller. A Parallax creation from the silicon on up, the Propeller chip's unique architecture and languages will change the way you think about embedded system design. The Propeller chip gives programmers both the power of true multi-processing and deterministic control over the entire system.

Each of the Propeller chip's processors, called cogs, can operate simultaneously, both independently and cooperatively with other cogs, sharing access to global memory and the system clock in a round-robin fashion through a central hub. Each cog has access to all 32 I/O pins, with pin states being tracked in its own input, output and direction registers. Each cog also has its own memory, 2 counter modules, and a video generator module capable of producing NTSC, PAL & VGA signals. Propeller Specifications:

- Languages: Spin (native, object-based), Assembly (native low-level), C/C++ (via PropGCC).

- Power Requirements: 3.3 VDC.
- Operating Temperature: -55 to +125 degrees C.
- Processors (Cogs): 8.
- I/O Pins: 32 CMOS.
- External Clock Speed: DC to 80 MHz.
- Internal RC Oscillator: ~12 MHz or ~20 kHz.
- Execution Speed: 0 to 160 MIPS (20 MIPS/cog).
- Global ROM/RAM: 32768/32768 bytes.
- Cog RAM: 512 x 32 bits/cog.

The Propeller is used in many industries including manufacturing, process control, robotics, automotive and communications. Hobbyists and engineers alike are finding new uses for this powerful microcontroller every day. The Propeller is a good choice over other microcontrollers when a low system part count is desirable due to its ability to provide direct video output and an easy interface to external peripherals such as keyboard, mouse and VGA monitor. Pre-written objects to support many types of hardware also make it an attractive option. All of this plus low cost and a powerful, yet easy language are hard to beat in a world where microcontrollers come in so many flavors that it's hard to make a choice.

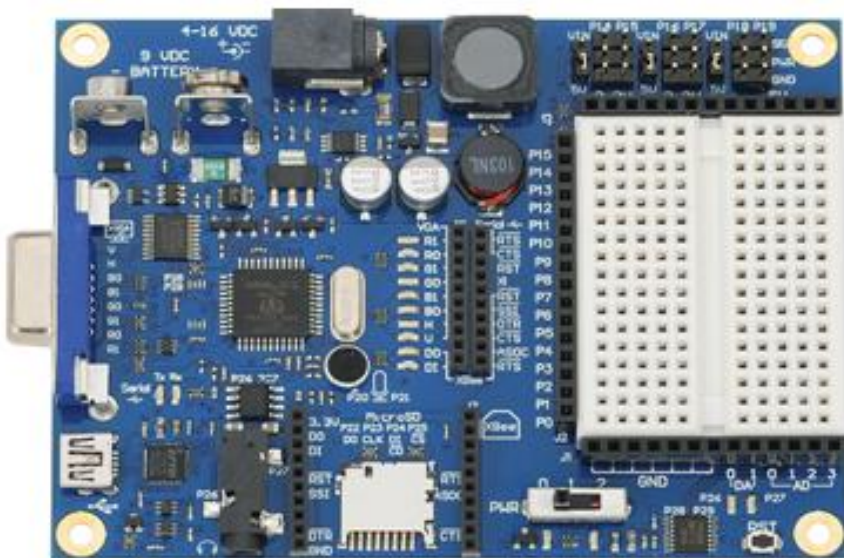
The Propeller chip is a multicore microcontroller that is programmable in high-level languages (Spin™ and C) as well as a low-level (Propeller assembly) language. Application development is simplified by using the set of pre-built objects for video (NTSC/PAL/VGA), mice, keyboards, LCDs, stepper motors and sensors. Propeller is easily connected to your computer's serial or USB port for programming using our Prop Plug. The Propeller chip can run on its own with a 3.3-volt power supply, internal clock, and with its internal RAM for code storage. Add an external EEPROM for non-volatile code storage and an external clock source for accurate timing.

The Propeller Tool Software is the primary development environment for Propeller programming in Spin and Assembly Language. It includes many features to facilitate organized development of object-based applications: multi-file editing, code and document comments, color-coded blocks, keyword highlighting, and multiple window and monitor support aid in rapid code development. We can use the board such as Propeller Robot board or Propeller

Board of Education for learning Propeller easily for robotics as shown in figure 2.1.



(a)



(b)

Figure 2.1 Propeller Chip P8X32A in LQFP package (a) and Propeller Board of Education (b).

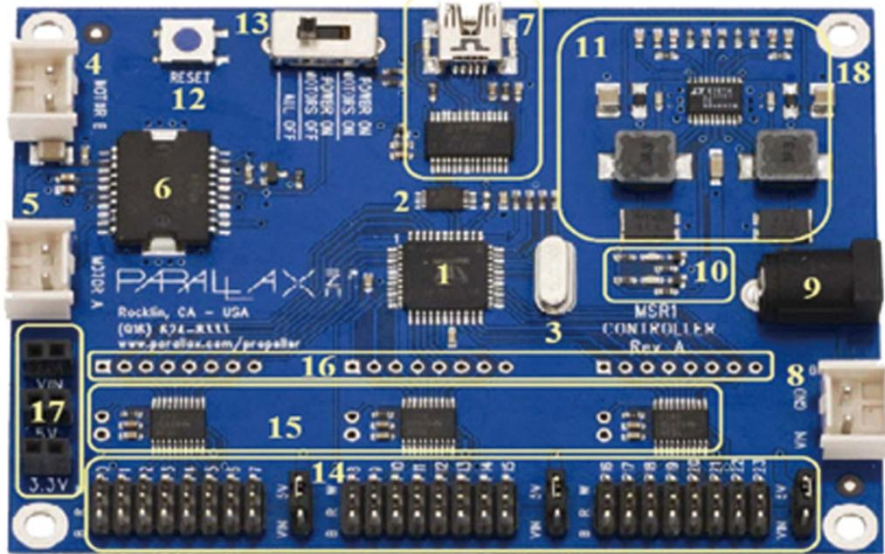
Table 2.1 pins description of propeller chip.

Pin Name	Direction	Description
		General purpose I/O Port A. Can source/sink 40 mA each at 3.3 VDC. Logic threshold is $\approx 1/2$ VDD; 1.65 VDC @ 3.3 VDC.
PO-P31	I/O	The pins shown below have a special purpose upon power-up/reset but are general purpose I/O afterwards. P28 – 12C SCL connection to optional, external EEPROM. P29 – 12C SDA connection to optional, external EEPROM. P30 – Serial Tx to host. P31 – Serial Rx from host.
VDD	---	3.3 volt power (2.7 – 3.3 VDC).
VSS	---	Ground.
BOEn	I	Brown Out Enable (active low). Must be connected to either VDD or VSS. If low, RESn becomes a weak output (delivering VDD through 5 K Ω) for monitoring purposes but can still be driven low to cause reset. If high, RESn is CMOS input with Schmitt Trigger.
RESn	I/O	Reset (active low). When low, resets the Propeller chip: all cogs disabled and I/O pins floating. Propeller restarts 50 ms after RESn transitions from low to high.
XI	I	Crystal Input. Can be connected to output of crystal/oscillator pack (with XO left disconnected), or to one leg of crystal (with XO connected to other leg of crystal or resonator) depending on CLK Register settings. No external resistors or capacitors are required.
XO	O	Crystal Output. Provides feedback for an external crystal, or may be left disconnected depending on CLK Register settings. No external resistors or capacitors are required.

The Propeller 2 is a whole-system, high-speed multicore chip for future embedded applications requiring real-time parallel control. Production customers asked for features that are now standard in Propeller 2: A/D, code protect, large RAM with freedom to download a C kernel or Spin interpreter during program. With easy coding for video (VGA, composite and component for HD), human interface devices, sensors and output devices, the Propeller 2 is

effective for quick prototype and production projects with limited time to market.

The Stingray robot from Parallax Inc. provides a mid-size platform for a wide range of robotics projects and experiments. The Propeller Robot Control Board is the brains of the system providing a multiprocessor control system capable of performing multiple tasks at the same time. The Propeller chip provides eight 32-bit processors each with two counters, its own 2 KB local memory and 32 KB shared memory. This makes the Propeller a perfect choice for advanced robotics and the Stingray robot. The board use is Propeller Robot Board complete with the USB Programmer, 64KB EEPROM AT24C512 and DC motor driver 7.2V as shown below:



- | | | |
|---------------------------|---------------------------------|---------------------------------|
| 1. Propeller P8X32A-Q44 | 7. USB Connector | 13. 3-Position Power Switch |
| 2. 64 KB EEPROM | 8. Battery Pack Socket | 14. Headers to I/O PO-P23 |
| 3. Socketed 5 MHz Crystal | 9. Barrel Jack, 2.1 mm | 15. Voltage Translators for I/O |
| 4. Right Motor Socket | 10. Power Good/Fail LEDs | 16. Direct I/O Access Vias |
| 5. Left Motor Socket | 11. Dual Switching Power Supply | 17. VIN, 5V, 3.3V Headers |
| 6. L6205 H-Bridge Driver | 12. Reset Button | 18. Ground/Mounting Vias |

Figure 2.2 Propeller Robot Control Board and the pins.

The general picture of the robot's assembly to produce differential wheeled robot models as shown below, and a complete description of the assembly can be read from manual of this robot:

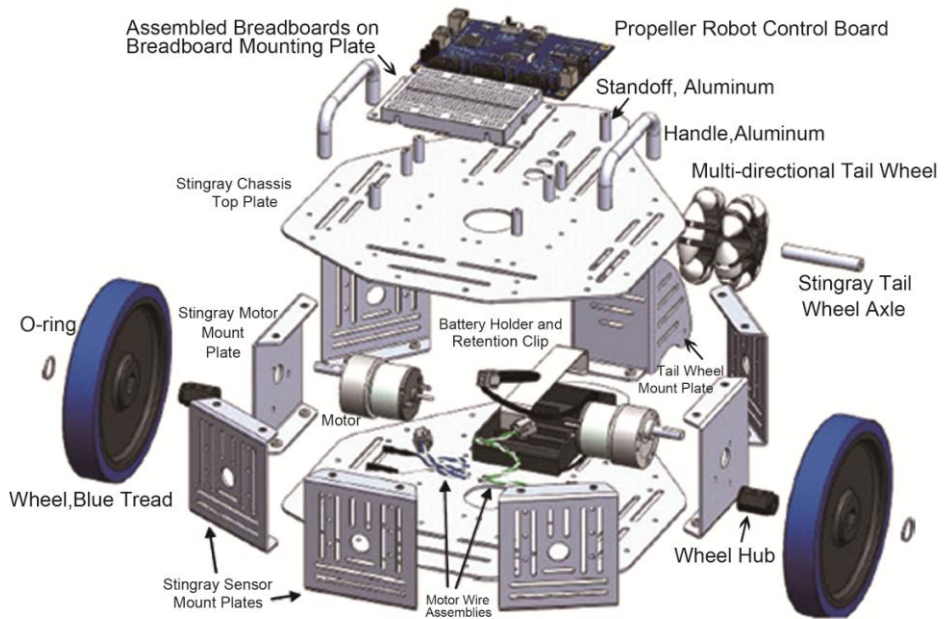


Figure 2.3 The general description of assembling the body, motors and the controller of the robot.

Programming the Propeller

We need USB/Serial programmer to program this chip, note that the connections to the external oscillator and EEPROM, which are enclosed in dashed lines, are optional as shown in figure 2.4 or figure 2.5 for serial programmer:

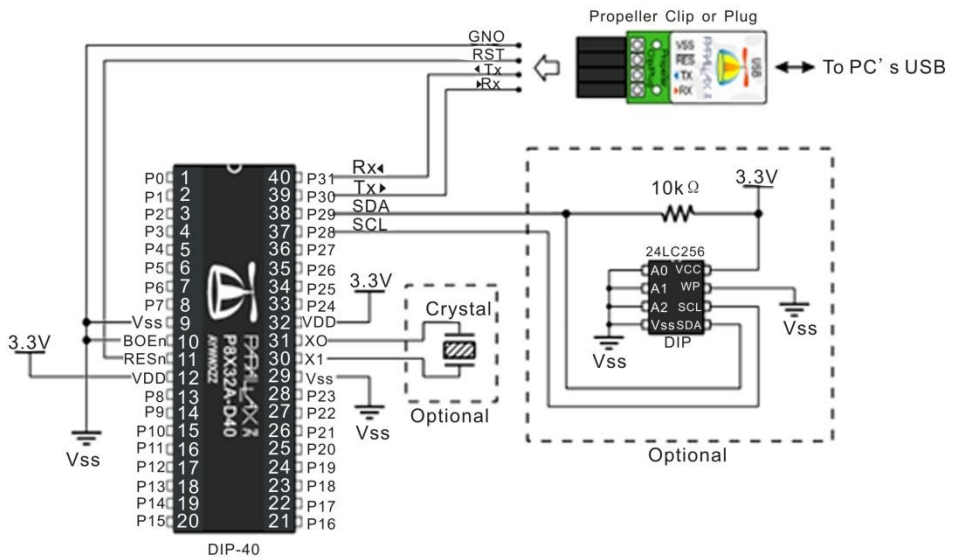


Figure 2.4 The minimum system of Propeller DIP-40 and the programmer.

The cheapest programmer for Propeller show below:

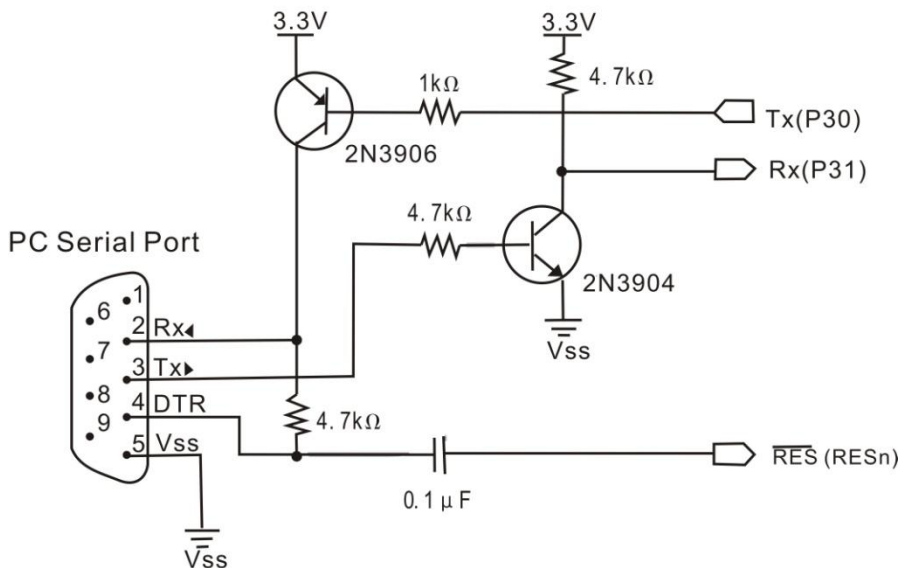


Figure 2.5 Schematic of serial programmer for Propeller.

After installing the Propeller tool software, the codes can be uploaded to the chip by pressing F11 as shown below:

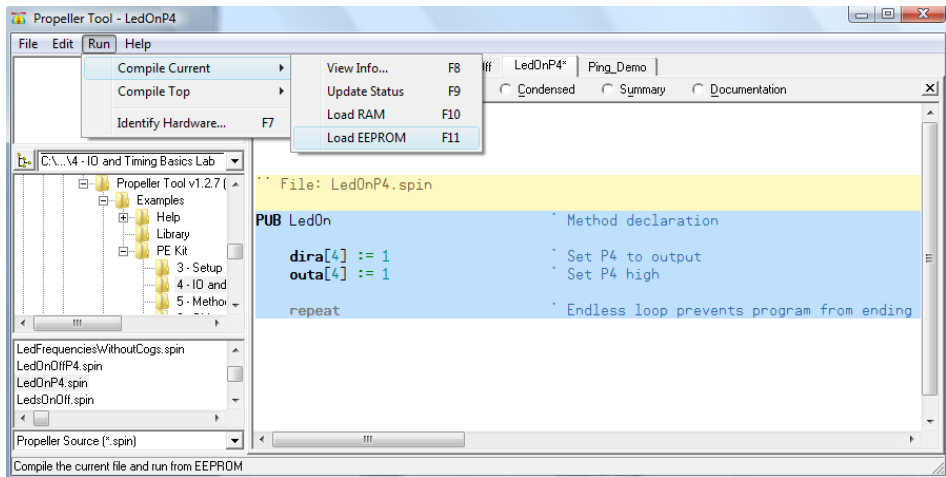


Figure 2.6 Programming the chip.

For basic experiments, we will try to program the LED lights on / off as well as receive input from switches. Install LED lights and switches on the protoboard are provided on the controller board as follows:

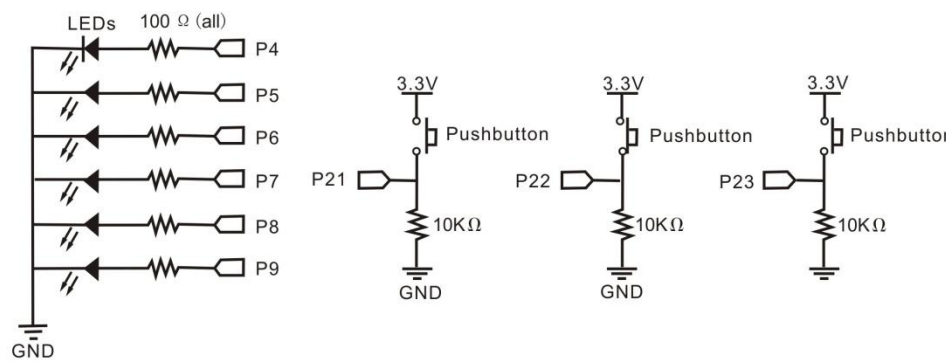


Figure 2.7 Schematic for basic testing of Propeller.

Here's an example of making light LED on / off at pin 4, save the file names LEDOnOffP4.spin by pressing F11, make sure the board Propeller detected on the USB port of your computer:

File: LEDOnOffP4.spin

```

PUB LedOnOff
  dira[4] := 1                      ' P4 → output

  repeat
    outa[4] := 1                    ' P4 → on
    waitcnt(clkfreq/4 + cnt)        ' delay
    outa[4] := 0                    ' P4 → off
    waitcnt(clkfreq/4 + cnt)

```



Figure 2.8 The USB connector and FTDI 232RL chip successfully detected the microcontroller.

Use the LEDs, resistors and pluggable wires to create the circuit shown in schematic below on the breadboard. The pluggable wires will jumper to the breadboard to make the I/O and ground connections from the control board locations shown below. Ground can be obtained from the bottom row of pins (marked B) on the I/O headers. P0 and P1 are picked up from the top row (marked W) and are indicated on the silkscreen on the control board. Power is obtained from the center row (marked R) and its voltage is set by the jumper immediately to the right of that group of headers.

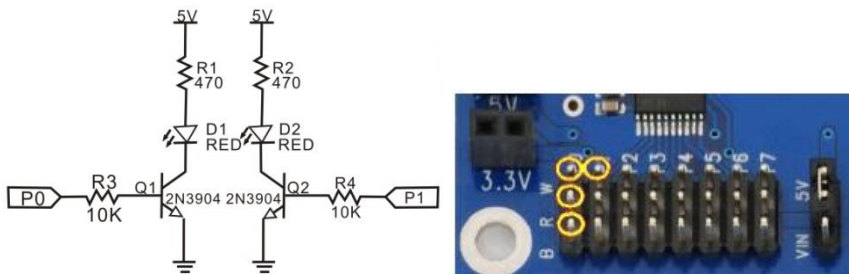


Figure 2.9 The schematic for using standard transistor.

The code for testing the transistor for driving LED shown below:

File: LED_Test.spin

```
CON
_xinfreq = 5_000_000      ' External Crystal frequency
_clokmode = xtall + pll16x ' Enabled external crystal and PLL
x16

PUB Main
  Dira[1..0] := %11          ' Set P0 and P1 to output

  Repeat
    Out[0] := 1              ' P0 HIGH
    Outa[1] := 0            ' P1 LOW
    Waitcnt (clkfreq/2 + cnt ) ' Delay ½ clock frequency (1/2
detik)
    Out[0] := 0              ' P0 LOW
    Outa[1] := 1            ' P1 HIGH
    Waitcnt (clkfreq/2 + cnt ) ' Delay ½ clock frequency (1/2
detik)
```

The next example is made in P6 LED lights on / off dependent input from P21, see what the output is generated when the switch is pressed in P21.

File: ButtonToLed.spin

```
PUB ButtonLED      ' Pushbutton/LED Method
  dira[6] := 1      ' P6 → output
  dira[21] := 0     ' P21 → input
  repeat           ' Endless loop
    outa[6] := ina[21] ' Copy P21 input to P6 ouput
```

That is a basic example of programming using the Propeller chip, you have to try other basic programming lies in the examples folder and library in the Propeller Tool program.

Exercises

- 1) Describe and compare features of some popular microcontrollers.
- 2) Design a minimum system for mobile robot using Propeller chip.

- 3) Create a program for Running LED using Propeller.
- 4) Create a program to control 2 DC Motors with an IC Driver L298 using switch.

Reference

- [1] Parallax.com.

