International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 2 Issue 4, April-2013, pp: (106-114), Available online at: www.erpublications.com

Development of Laboratory Scale Self-Balancing Prototype Robot

M. K. A. Ahamed Khan[1], I. Elamvazuthi[2], P. Vasant[3], T. Ganesan[4], R. A. Amin[5], S. Parasuraman[6]
[1]Senior Lecturer, Universiti Industri Selangor, Kampus Bestari Jaya, Malaysia
[2345]Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Malaysia
[6]Monash university, Malaysia
[1]khan@unisel.edu.my

Abstract: There are many projects on building self-balancing robots - ranging from simple DIY analog balancing bots to sophisticated self-balancing scooters. Self-balancing electro-mechanical systems have various uses, including the ability to showcase the computational performance of new and emerging embedded processors. The project would primarily focus on the software implementation of the algorithm by incorporating a physical model of the robot and the available sensors onboard the PIC controller. The software itself is intended to be portable and can be used on any self-balancing platform based on the controller by tuning a few parameters. This book chapter presents the development of a laboratory scale self-balancing prototype robot. It addresses the balancing mechanism issue based on the theories of inverted pendulum and rotational inertia. It comprises of two major sub-systems, i.e., mechanical and control sub-systems. The implementation of the mechanical and control sub-systems is described in detail.

Keywords: self-balancing robot, mechanical, control, inverted pendulum.

1. Introduction

Over the last decade, the research on self-balancing robot has gained interest amongst the robotics enthusiasts as well as the robotics laboratory around the world [1]. There are considerable research efforts towards building a robust self-balancing robot [2-7]. The application of self-balancing robot is unlimited where it can serve the community and encounter the problems that many have raised in the industrial and service sectors. For example, a motorized wheelchair utilizing this technology would give the operator greater maneuverability and thus access to place most able-bodied people take for granted [8].

As compared to four-wheels trolley, having a platform with only two wheels that is able to balance itself and move around is advantageous as it possibly maneuvers better, move on either flat or sloppy surfaces, and to further enhances, apply for carrying limitless tasks, hence benefit the community. Yet, the main challenges to this possibility is how to have a two wheels robot that is able to stand upright and balance itself, furthermore carry object from one point to other point efficiently. This is where the significant of this project comes from which is to build a self-balancing robot [9].

The main objective for this project is to build a self-balancing robot. The robot is comprised platforms with two wheels and equipped with successful balancing mechanism. The self-balancing robot should be able to stand upright and balance itself, move from one point to other point as pre-programmed and carry an object from one point to the other.

This book chapter is organized as follows: Section II presents the self-balancing robot dynamics. Section III provides the system design. Section IV discusses the implementation details and finally, section V is the conclusions.

2. Self-Balancing Robot Dynamics

The derivation of dynamics of self-balancing robot is given in the following based on Figure 1. For detail derivation, see [10].

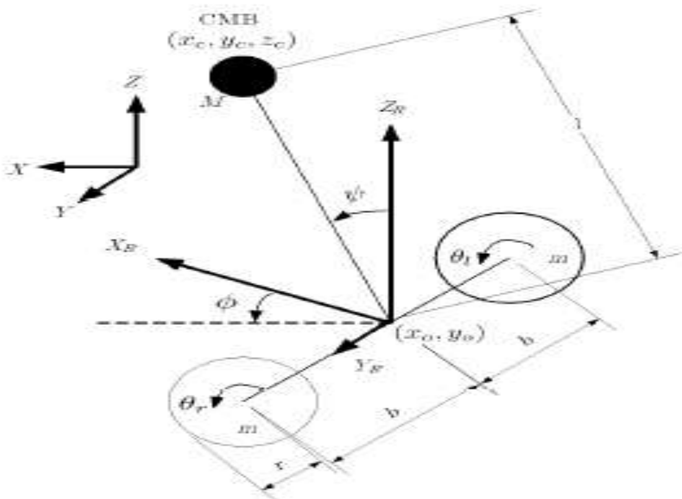


Figure 1: Dynamic Model of Self-Balancing Robot [10]

The system parameters based on Figure 1 is given in Table 1.

Table 1 System Parameters

r	Radius of wheel
m	Mass of a wheel
J_c	Moment of inertia of body
J_y	Moment of inertia of the motor and the wheel about the y axis
J_d	Moment of inertia of the motor and the wheels about their diameter
b_r, b_l, b_v, b_w	Damping coefficients of the robot
θ	Rotational angle of the wheels
ψ	Tilt angle of CMB
ω	The angular velocity of the origin
ϕ	Steering angle about the origin
M	Mass of the CMB
l	Distance between CMB and origin
$2b$	Distance between wheels
b_m, J_m, R_a, L_a, k_m	Motor parameters

Given the kinetic energy of the wheels and Centre Mass of Body (CMB), the damping energy, and potential energy, as K_ω, K_c, β, U_c respectively, we obtain,

$$K_\omega = \frac{1}{2}(mr^2 + J_y)(\dot{\theta}_r^2 + \dot{\theta}_l^2) + J_d \dot{\phi}^2 \qquad (1)$$

$$K_c = \frac{1}{2} v_c^T M v_c + \frac{1}{2}\omega_c^T J_c \omega_c \qquad (2)$$

$$\beta = \frac{1}{2} b_r(\dot{\theta}_r - \dot{\psi}_l)^2 + \frac{1}{2} b_l(\dot{\theta}_l - \dot{\psi}_l)^2 + \frac{1}{2} v_c^T b_v v_c + \frac{1}{2}\omega_c^T b_c \omega_c \qquad (3)$$

$$U_c = Mgl\cos\psi \qquad (4)$$

The robot is powered by two motors. The controlled variables of the model are the position and orientation of the mobile robot, while the control variables are the angular velocities of the left wheel and the right wheel.

3. System Design

The laboratory prototype self-balancing robot is divided into two major sub-systems, i.e., mechanical and control sub-systems.

A. Mechanical Sub-system

The mechanical sub-system comprises of three round platforms arranged vertically, two wheels at the bottom of each side, with circuitry, servo motors and batteries attached to the structure as illustrated in Figure 2.

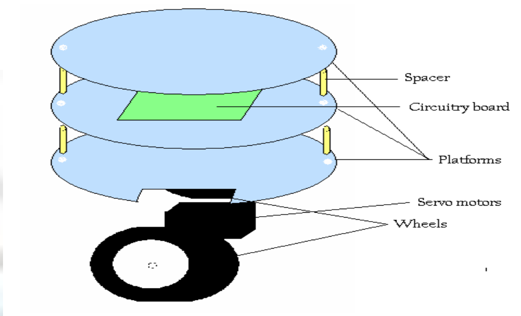


Figure 2: Mechanical sub-system

The base area for the mechanical sub-system is designed to be reasonably wide. Theoretically, the wider the base area is the more stable the structure would be. Even so, the structure for this is not required to be perfectly stable without the balancing mechanism. The structure needed to wobble or tilt at significant for the accelerometer, Memsic2125 to sense it easily, or else the balancing mechanism may not work as expected. At the same time, the structure should not be in the state of not very stable or else it would difficult for it to balance. Hence, the base is determined as 150mm in diameter in accordance to the dimension of the platforms, as shown in Figure 3. Note that, the size of the wheels, which is 50mm in diameter, is taken into consideration in determining the base area. Also note that, the material of the platforms is 2.5mm Perspex. The purpose of this choice is to keep the total weight of the structure at a range that the servo motors can operate at, which is 8 kg at maximum.

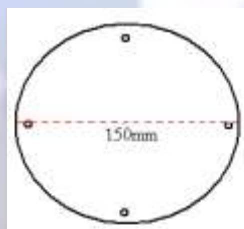


Figure 3: Dimension of the platform

Three round platforms are used in this project. The first platform which is the base platform is a round shape with two rectangle shapes cut out at each side, as shown in Figure 4. Those rectangles are meant for the wheels.

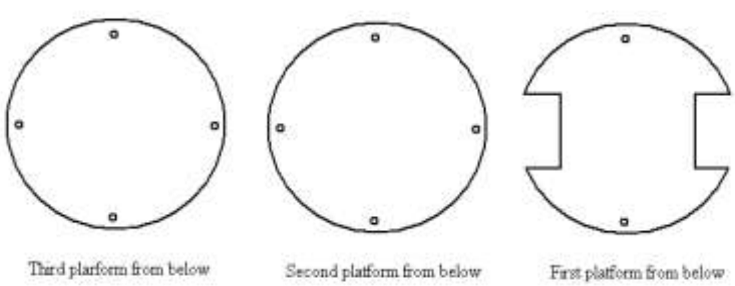


Figure 4: Platforms for structure

The dimension of the wheels for this project is chosen to be reasonably large also depends on its availability in the market. Hence the dimension of the wheels is 50mm diameter and 25mm wide. The wheels are Tamiya manufactured.

B. Control Sub-system

The microcontroller chip used is AtMega16, the developed circuit schematic is as shown in Figure 5. This circuitry comprises power, reset, clock, input, output connections and in-circuit programming.

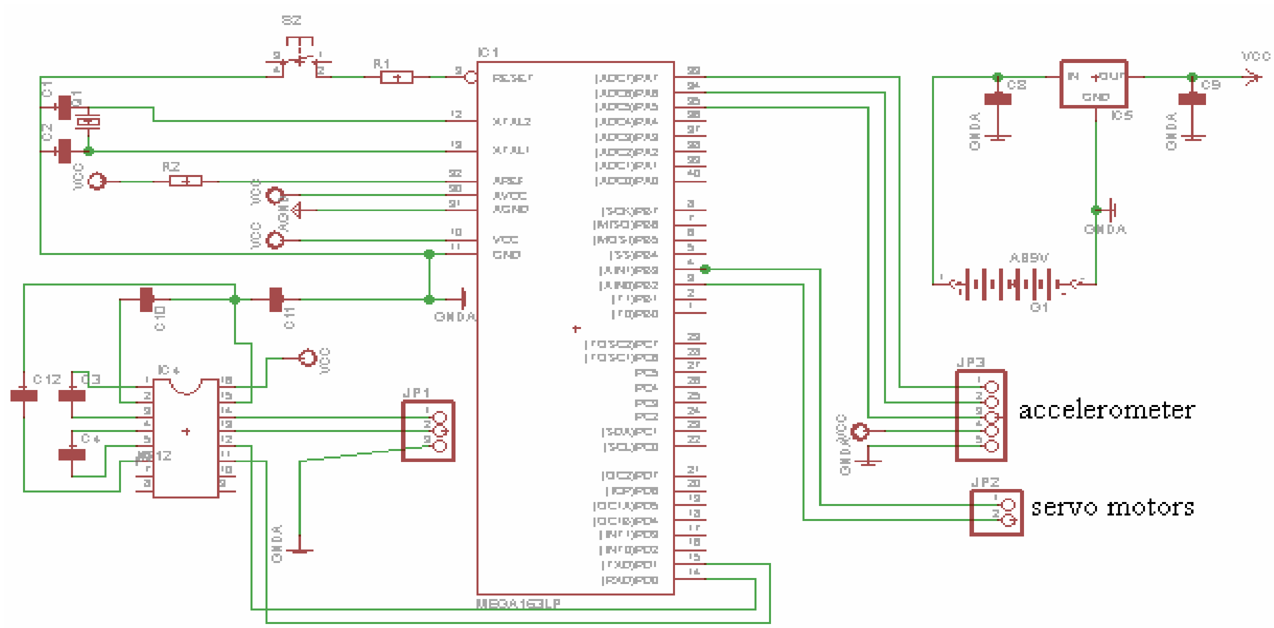


Figure 5: Schematic for circuitry

The purpose of power circuit is to regulate the battery's voltage from 12V to 5V as appropriate for this circuit. The reset circuit is to reset the operation of AtMega16, whilst, the clock circuit is to provide clock cycle for the AtMega16 so that it could operate faster. It should be noted that connecting to high frequency of crystal clock may not assure that the microcontroller chip works as desired, in a worse case may damage the chip. Therefore, a proper frequency of crystal clock has to be ascertained depending on the type of microcontroller chip. Accelerometer pin-outs are for the Memsic2125 connections and servo motors pin-outs are for servo motors connections. The purpose to have in-circuit programming is to eliminate the need to detach and reattach AtMega16 for programming. This would also eliminate the possibility of damaging the pins of AtMega16.

The circuit design is transferred to PCB (printed circuit board) design using Eagle software. The design for the PCB is as shown in Figure 6.

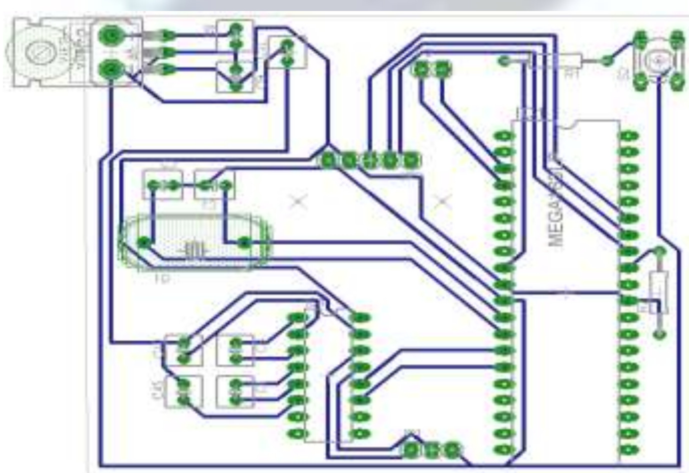


Figure 6: PCB layout for circuitr

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 2 Issue 4, April-2013, pp: (106-114), Available online at: www.erpublications.com

4. Implementation Details

The mechanical structure of the robot is made of a combination of Aluminium and Perspex. The control subsystem of all the robots is comprised of a PIC Microcontroller Board, DC Brushless motor, brush motor, IR sensors, solenoids, and limit switches. The developed laboratory scale self-balancing prototype robot is shown in Figure 7.

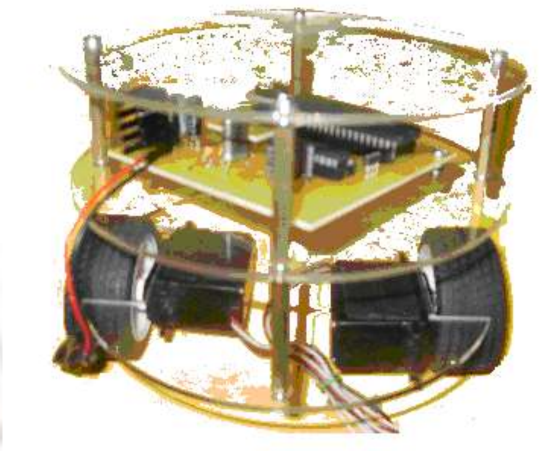


Figure 7: Self-Balancing Robot

Figure 8 shows the control sub-system. The main function of the control system is to generate electrical signals to effect motion based on input given by the control software. Considering the requirements of the self-balancing robot, it was decided that the simplest and most cost effective way of achieving the sequence control was to use a AtMega16 microcontroller board which acts as the heart of the robot.

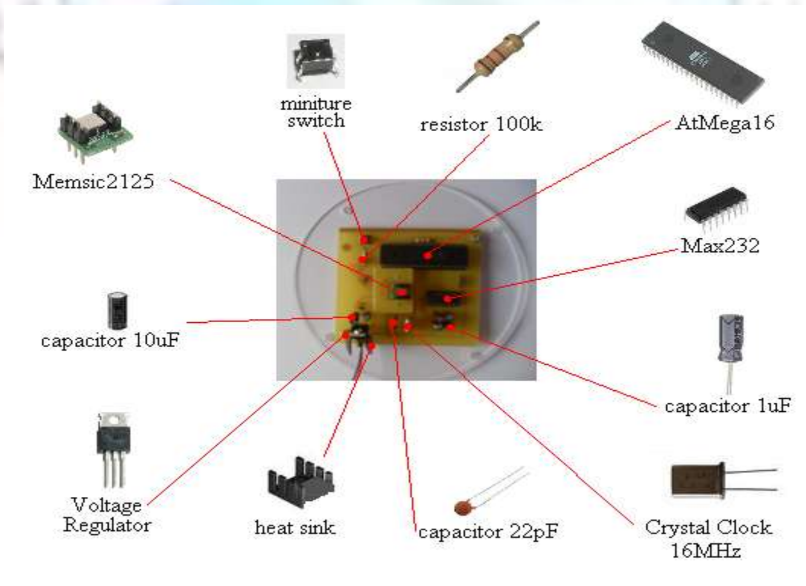


Figure 8: Control Sub-system

The rest of the components are carefully soldered unto the PCB board.

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, theATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 2 Issue 4, April-2013, pp: (106-114), Available online at: www.erpublications.com

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

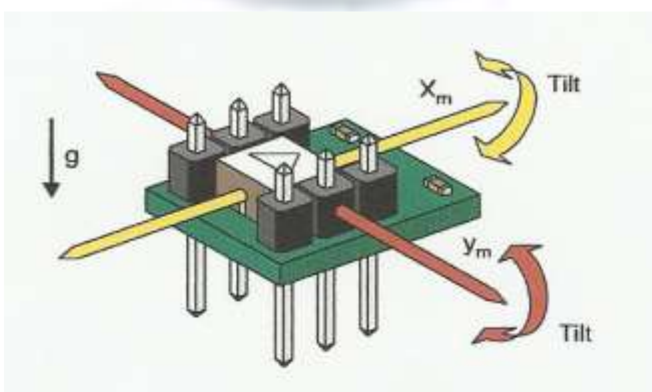
The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only),a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run. The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications. The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

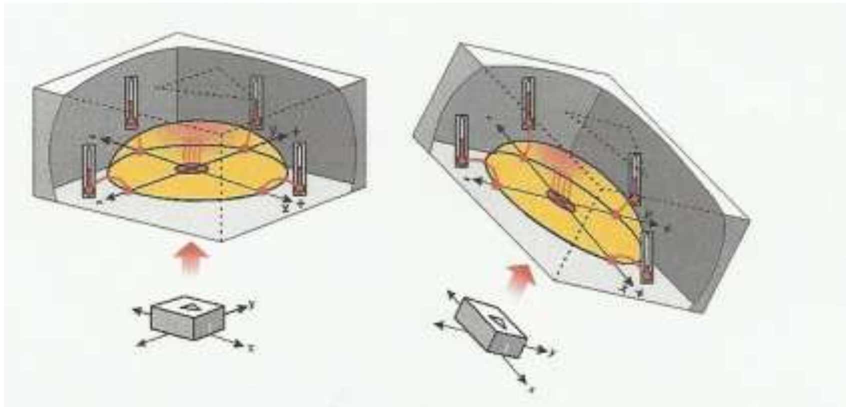
5. Memsic2125

Theory of Operation

The MX2125 has a chamber of gas with a heating element in the center and four temperature sensors around its edge. When the accelerometer is level, the hot gas pocket rises to the top-center of the chamber, and all the sensors will measure the same temperature.
By tilting the accelerometer, the hot gas will collect closer to some of temperature sensors. By comparing the sensor temperatures, both static acceleration (gravity and tilt) and dynamic acceleration (like taking a ride in a car) can be detected. The MX2125 converts the temperature measurements into signals (pulse durations) that are easy for microcontrollers to measure and decipher.



Memsic 2125 Dual-Axis Accelerometer

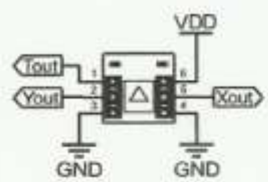


Operational Functions of Memsic 2125 Dual-Axis Accelerometer

Pin Definitions

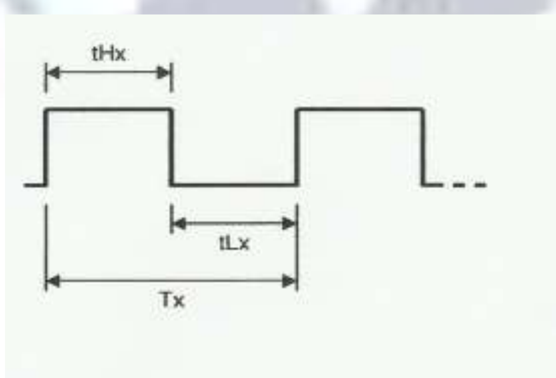
For Memsic MXD2125GL pin ratings, see the manufacturer's datasheet posted on the 28017 product page at www.parallax.com.

Pin	Name	Function
1	Tout	Temperature Out
2	Yout	Y-axis PWM output
3	GND	Ground -> 0 V
4	GND	Ground -> 0 V
5	Xout	X-axis PWM output
6	VDD	Input voltage: +3.3 to +5 VDC



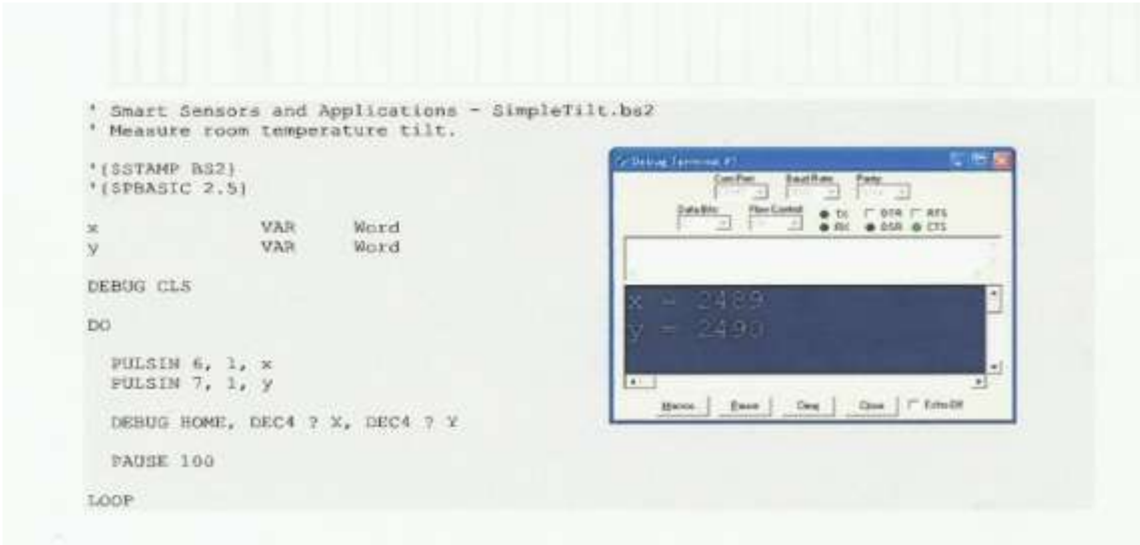
Communication Protocol

Each axis has a 100 Hz PWM duty cycle output in which acceleration is proportional to the ratio tHx/Tx. In practice, we have found that Tx is consistent so reliable results can be achieved by measuring only the duration of tHx. This is easy to accomplish with the BASIC Stamp PULSIN command or with the Propeller chip's counter modules. With Vdd = 5V, 50% duty cycle corresponds to 0 g, but this will vary with each individual unit within a range of 48.7% to 51.3%. This zero offset may be different when using Vdd = 3.3 V.



Communication Protocol signal

The program below, SimpleTilt.bs2, simply measures the pulse width, that is, the duration of tHx, for each axis. The raw values are displayed in the BASIC Stamp Editor's Debug Terminal. If you run the program, then tilt the accelerometer, you should see the values for each axis change.

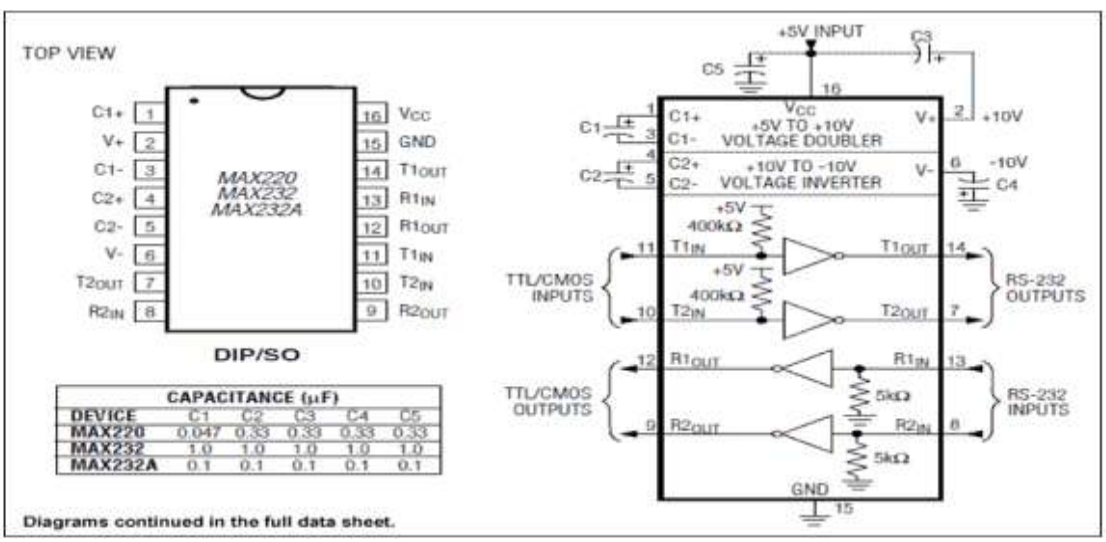


6. MAX232

The **MAX232** is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case. The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 VTTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors – 0.1 μF in place of the 1.0 μF capacitors used with the original device.[1] The newer MAX3232 is also backwards compatible, but operates at a broader voltage range, from 3 to 5.5 V. [2]
Pin to pin compatible: ICL232, ST232, ADM232, HIN232 (much cheaper alternatives).



Programming the microcontroller is necessary to perform the desired application of the robot. These codes enable the robot to do its explicit function, designed by the programmer depending on the tasks that are needed to be completed. The functional diagram of the robot is shown in Figure 9.

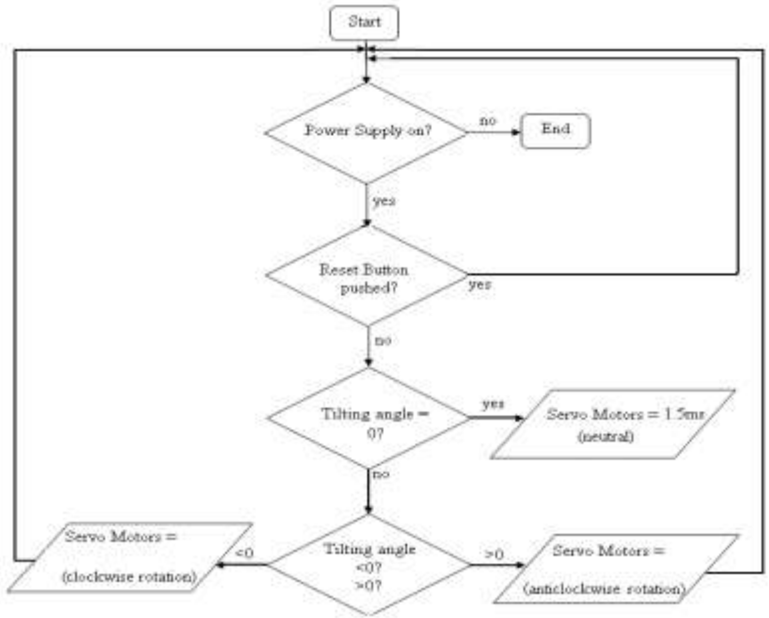


Figure 9 Functional diagram

From Figure 9, it can be seen that initially, it ensures whether the power supply is turned on or off. Then, it senses for reset button. If the reset is pressed, the flow continues to the input data from Memsic2125. This would then continue in a loop. A series of tests were carried out to determine the robustness of the algorithms and it performed well the repeatability reaching 73%.

7. Conclusion

A laboratory scale self-balancing robot was developed and tested. The robot was able to perform the desired tasks effectively. Further research should study the robustness of the control strategy with additional incorporation of IR sensors or gyroscope for better maneuverability and control.

8. Acknowledgements

The authors are grateful to UNISEL and UTP for supporting this work.

9. References

[1]. Hu, J. S, 2009, "Self-balancing Control and Manipulation of a Glove Puppet Robot on a Two-wheel Mobile Platform", International Conference on Intelligent Robots and Systems, St. Louis, USA.
[2]. Ha, Y. and Yuta, S. Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot. Robot. Autonomous Systems, 1996, 17, 65 – 80.
[3]. Kumagai, M. and Ochiai, T., "Development of a Robot Balancing on a Ball," Proc, ICCAS 2008, 2008, pp 433 – 438.
[4]. http://www.geology.smu.edu/~dpa-www/robo/nbot/.
[5]. Segway, LLC, www.segway.com
[6]. Thomas Braunl, Embedded Robotics, Berlin Heildelberg: Springer – Verlag, 2006, ch.9.
[7]. Ong Yin Chee, Mohamad Shukri b. Zainal Abidin, "Design And Development of Two Wheeled Autonomous Balancing Robot", 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, 27-28 June 2006.
[8]. N. M. Abdul Ghani, N. I. Mat Yatim, N. A. Azmi, Comparative Assessment for Two Wheels Inverted Pendulum Mobile Robot Using Robust Control", International Conference on Control, Automation and System 2010, KINTEX, Gyeonggi-do, Korea, Oct 27-30, 2010.
[9]. Hua Sun, Haixu Zhou, Xiang Li, Yanhui Wei and Xiao Li, "Design of Two-Wheel Self-Balanced Electric Vehicle based on MEMS", Proceedings of the 2009 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, Shenzhen, China, January 5-8, 2009.
[10]. Araghi, M.H. and Kermani, M.R., "Computer-Aided System Design for Educational Purposes: An Autonomous Self-Balancing Two-Wheeled Inverted Pendulum Robot", IEEE, CCECE 2011.