CERTIFICATE

This is to certify that the project report entitled

"Remote Access Ultrasonic Anemometer"

Submitted by

Aneesh N. Abhyankar	Exam No. B80233005
Aniket S. Parandkar	Exam No. B80233136
Viraj V. Chitale	Exam No. B80233035

Is a bonafide work carried out by them under the supervision of Prof. Dr. R. S. Kawitkar and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Electronics and Telecommunication)

This seminar report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. Dr. R. S. Kawitkar	Prof. Dr. M. B. Mali	Prof. Dr. S. D. Lokhande
Guide	Head	Principal
Department of E&TC	Department of E&TC	Sinhgad College of Engineering

Place: Pune

Date: / / 2015

CERTIFICATE FROM INDUSTRY

भारत सरकार
भारत मौसम विज्ञान विभाग

मौसम विज्ञान के उप–महानिदेशक
(सतह उपकरण) का कार्यालय
शिवाजीनगर, पुणे-411 005, भारत

GOVERNMENT OF INDIA
INDIA METEOROLOGICAL DEPARTMENT

Office of the
Deputy Director General of Meteorology
(Surface Instruments)
Shivajinagar, Pune 411 005, INDIA

ISO 9001:2008 CERTIFIED

No. No. I/EL-333 (Project)/ Date 13th August 2014.

To,

Sinhgad college of Engineering
S.No.44/1, Vadgaon (BK),
Off Sinhgad Road, Pune – 411041.

Sub: Students project sponsorship

Ref: Your letter no. SCOE/ETC/PRJ-RCO/2014-15/487 dt.03.07.2014.

Sir,

The following students of final year B.E are permitted to undertake a project entitled "Remote Access Anemometer" in this office under the guidance of Shri K.N.Mohan, Scientist 'E' during the academic year 2014 – 15.

1. Viraj Chitale
2. Aneesh Abhyankar
3. Aniket Parandkar

However the cost involved in purchase of required components for design & development of the project is to be borne by the students.

Thanking you.

Yours faithfully,

13/8/14

(P.N.Mohanan)
Scientist 'E'
for DDGM (SI)

Telephone +91-20-25572424, 25572428, 25572426, 25572429, 25572412, 25572404
FAX : +91-20-2552 1529 • E-mail : ddgmsi@imd.gov.in

ACKNOWLEDGEMENT

Any accomplishment requires the effort of many people and this work is no different. We find great pleasure in expressing our deep sense of gratitude towards all those who have made it possible for us to complete this project with success.

We would like to express our trust and sincerest gratitude to **Prof. Dr. R. S. Kawitkar**, our guide for his dynamic and valuable guidance and keen interest in our project work. We are grateful to him for his constant encouragement in the fulfillment of the project work. The project has been a result of combined efforts put in by our guide and us. He has been a strong moral support to us throughout this project. We would also like to thank him for providing us with all the necessary infrastructure and facilities to complete the project. We would also like to extend our sincere gratitude to **Shri P. N. Mohanan** and **Shri K. N. Mohan** at Indian Meteorological Department, Pune who helped us proactively in completing this project.

We also extend our sincere thanks to the Head of Department of Electronics and Telecommunication Engineering, **Dr. M. B. Mali** and all the **staff members** who extended the preparatory steps of this project and seminar.

We would also like to thank all our friends and our parents who knowingly assisted us with their valuable suggestions and support and we are grateful for their assistance. We would also like to thank those who were directly or indirectly guiding us.

ANEESH N. ABHYANKAR

ANIKET S. PARANDKAR

VIRAJ V. CHITALE

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

An anemometer is a device used for measuring wind speed and wind direction. This project will be a significant improvement over a cup and vane anemometer, which are mechanical devices. A cup anemometer has four round cups, in pairs mounted opposite to each other on a rotating shaft. The speed of rotation determines the speed of the wind. The vane anemometer aligns itself with the wind direction. The ultrasonic anemometer has four firmly placed ultrasonic transceivers in place of the cups. Using changes in the speed of transmission of ultrasonic sound pulses between opposite transceivers, we can determine the wind speed as well as the wind direction. Now-a-days, the ultrasonic anemometers are mounted on the buildings and the data can be accessed by connecting them to computers via an RS232 / RS485 interface. Our objective is to introduce a microcontroller between the computer and the anemometer. This would help us access the data from the anemometer remotely. Hence the name remote access ultrasonic anemometer.

1.1 BACKGROUND

Wind is the natural motion of the air roughly parallel to the Earth's surface. It is caused by the unequal heating and cooling of the Earth and atmosphere by the sun, which produces differences in air pressure. As the atmosphere shifts air masses to equalize these differences wind is developed, tending to flow from areas of high pressure to areas of low pressure. Additional factors also come into play that can influence the wind speed and direction, such as Earth's rotation (Coriolis Effect), the condensation of water vapor, the formation of clouds, friction over land and water, and others.

Wind occurs at all scales - global winds (trade winds), upper level winds (jet streams), synoptic winds (resulting from the pressure differences of surface air masses), local (mesoscale) winds (such as gust fronts), and winds that develop because of geographical features (like sea breezes). Winds also occur on a much smaller scale, for example dust devils or tornadoes.

Wind observations are taken at a fixed location using two parameters: wind speed and wind direction. Referenced with respect to true North, the direction that the wind is flowing from is measured in degrees. It also described by the compass points it's flowing from (N, NE, NNE, etc.). Wind speed is a measurement of the speed of movement of the air, and is typically reported in miles per hour (mph) or kilometers per hour (kph). Reports for maritime and aeronautical operations may use those or knots (nautical miles per hour).

1.2 RELEVANCE

The measurement of wind speed is one of the most important factors in weather prediction. Wind is the movement of air caused by uneven heating of the earth's surface. It occurs in light breezes that are locally generated due to heating of an immediate landmass, to winds on a grand scale spanning continents caused by solar heating. Besides being used as part of a weather monitoring station there are many other situations where measurement and knowledge of the wind condition helps in decision-making such as pollution control, safety of tall structures, control of wind turbines, studies on the effects of wind on crops, maneuvering of ships and aircraft landing systems. Wind speed is measured with an anemometer; the word anemometer comes from the Greek *anemos* meaning wind, plus meter "anemometer"

In the wind energy field, there are two primary reasons we might want to measure wind speed:-

- To determine feasibility of wind power development at a site.
- As part of a wind turbine control system.

Essentially, we measure wind speed and direction to answer the question: "Is it worthwhile to turn the turbine into the wind and start it?"

Knowledge required for the design of wind turbines includes:-

- Wind speed and direction
- Inclination to the ground
- Turbulence levels
- General weather conditions – Temperature, Pressure, Humidity, etc.

1.3 PROJECT UNDERTAKEN

To design an ultrasonic anemometer and the corresponding circuitry to measure wind speed and direction in real time and transfer the data to the user over the Ethernet, thus making it a remote access device

The Ultrasonic Anemometer 2Dis designed to detect the horizontal components of wind speed and wind direction in two dimensions.

The Ultrasonic Anemometer 2Dconsists of 4 ultrasonic transducers, in pairs of 2 which are opposite each other at a distance of 200 mm. The two measurement paths thus formed are vertical to each other. The transformers act both as acoustic transmitters and acoustic receivers. The respective measurement paths and their measurement direction are selected via the electronic control. When a measurement starts, a sequence of 4 individual measurements in all 4 directions of the measurement paths is carried out at maximum possible speed. The measurement directions (acoustic propagation directions) rotate clockwise, first from south to north, then from west to east, from north to south and finally from east to west. The mean values are formed from the 4 individual measurements of the path directions and used for further calculations.

The speed of propagation of the sound in calm air is superposed by the speed components of an air flow in wind direction. A wind speed component in the direction of the propagation of the sound supports the speed of propagation, thus leading to an increase in the speed. A wind speed component opposite to the direction of propagation, on the contrary, leads to a reduction of the speed of propagation. The speed of propagation resulting from the superposition leads to different propagation times of the sound at different wind velocities and directions over a fixed measurement path. As the speed of sound is very dependent on the air temperature, the propagation time of the sound is measured on both of the measurement paths in both directions. In this way, the influence of the temperature-dependent speed of sound on the measurement result can be eliminated. By combining the two measuring paths which are at right angles to each other, one obtains the measurement results of the sum and the angle of the wind speed vector in the form of rectangular components. After the rectangular speed components have been measured, they are then transformed by the μ-processor of the anemometer into polar co-ordinates and output as sum and angle of wind speed.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 INTRODUCTION

An anemometer or wind meter is a device used for measuring wind speed, and is a common weather station instrument. The term is derived from the Greek word anemos, meaning wind, and is used to describe any air speed measurement instrument used in meteorology or aerodynamics. The first known description of an anemometer was given by Leon Battista Alberti around 1450. We have designed an ultrasonic anemometer to measure wind speed and direction using a microcontroller ATMega 328 and transfer the data via internet using Ethernet controller.

2.2 ANEMOMETERS

Anemometers can be divided into two classes: those that measure the wind's speed, and those that measure the wind's pressure; but as there is a close connection between the pressure and the speed, an anemometer designed for one will give information about both.

2.2.1. Cup anemometers

A simple type of anemometer was invented in 1845 by Dr. John Thomas Romney Robinson, of Armagh Observatory. It consisted of four hemispherical cups, each mounted on one end of four horizontal arms, which in turn were mounted at equal angles to each other on a vertical shaft. The air flow past the cups in any horizontal direction turned the shaft in a manner that was proportional to the wind speed. Therefore, counting the turns of the shaft over a set time period produced the average wind speed for a wide range of speeds. On an anemometer with four cups, it is easy to see that since the cups are arranged symmetrically on the end of the arms, the wind always has the hollow of one cup presented to it and is blowing on the back of the cup on the opposite end of the cross.

2.2.2. Vane anemometers

One of the other forms of mechanical velocity anemometer is the vane anemometer. It may be described as a windmill or a propeller anemometer. Contrary to the Robinson anemometer, where the axis of rotation is vertical, the axis on the vane anemometer must be parallel to the direction of the wind and therefore horizontal. Furthermore, since the wind varies

in direction and the axis has to follow its changes, a wind vane or some other contrivance to fulfill the same purpose must be employed.

A vane anemometer thus combines a propeller and a tail on the same axis to obtain accurate and precise wind speed and direction measurements from the same instrument. The speed of the fan is measured by a rev counter and converted to a wind speed by an electronic chip. Hence, volumetric flow rate may be calculated if the cross-sectional area is known.

2.2.3. Sonic anemometers

Sonic anemometers, first developed in the 1950s, use ultrasonic sound waves to measure wind velocity. They measure wind speed based on the time of flight of sonic pulses between pairs of transducers. Measurements from pairs of transducers can be combined to yield a measurement of velocity in 1-, 2-, or 3-dimensional flow.

Two-dimensional (wind speed and wind direction) sonic anemometers are used in applications such as weather stations, ship navigation, wind turbines, aviation and weather buoys. Three-dimensional sonic anemometers are widely used to measure gas emissions and ecosystem fluxes using the eddy covariance method when used with fast-response infrared gas analyzers or laser-based analyzers.

There are different technologies for two-dimensional wind sensor.

- Two ultrasounds paths: These sensors have 4 arms. These sensors are commonly used by the wind sensor manufactured because it's easier to develop compare to the three ultrasounds paths. The disadvantage of this type of sensor is that when the wind comes in the direction of an ultrasound path the flow around the arms introduces aerodynamic turbulences which perturb the measures and the sensor will lose in accuracy.
- Three ultrasounds paths: These sensors have 3 arms. These sensors are developed by some wind sensors manufactured because it gives one path redundancy of the measure which improve the sensor accuracy and minimize the aerodynamic turbulences.

2.3 MICROCONTROLLER

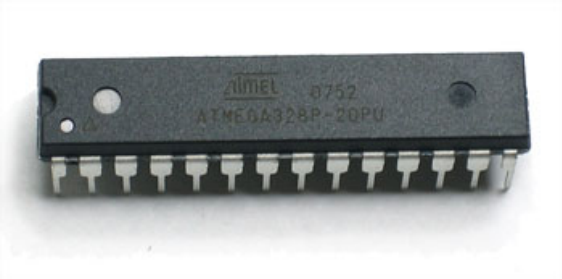


Figure 2.1 ATmega328P Microcontroller (Courtesy: www.atmel.com)

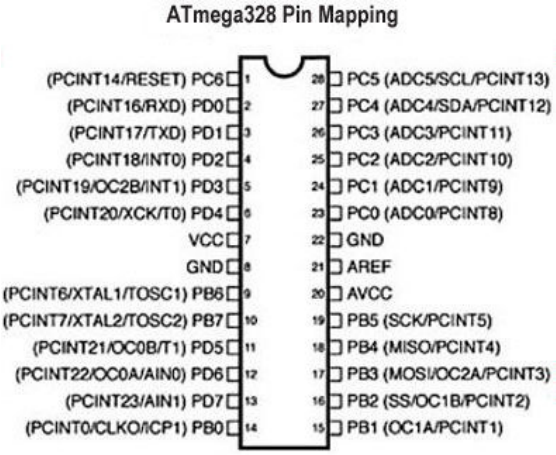


Figure 2.2 ATmega328P Pinout (Courtesy: www.instructables.com)

ATMEGA 328P - Features

1. High Performance, Low Power Atmel AVR 8-Bit Microcontroller Family.
2. Advanced RISC Architecture
    - 131 Powerful Instructions – Most Single Clock Cycle Execution
    - 32 x 8 General Purpose Working Registers
    - Fully Static Operation
    - Up to 20 MIPS Throughput at 20MHz
    - On-chip 2-cycle Multiple.

3. High Endurance Non-volatile Memory Segments
    - 4/8/16/32 Kbytes of In-System Self-Programmable Flash Memory
    - 256/512/1Kbytes EEPROM
    - 512/1K/2Kbytes Internal SRAM
    - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
    - Data Retention: 20 years at 85°C
    - Optional Boot Code Selection with Independent Lock Bits
4. Peripheral Features
    - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
    - One 16-bit Timer/Counter with Separate Prescaler and Compare Mode
    - Real Time Counter with Separate Oscillator
    - Six PWM Channels
    - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - 6-channel 10-bit ADC in PDIP Package
    - Programmable Serial USART
    - Master/Slave SPI Serial Interface
    - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
    - Programmable Watchdog Timer with Separate On-chip Oscillator
    - On-chip Analog Comparator
    - Interrupt and Wake-up on Pin Change
5. Special Microcontroller Features
    - Power-on Reset and Programmable Brown-out Detection
    - Internal Calibrated Oscillator
    - External and Internal Interrupt Sources
    - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
6. I/O and Packages
    - 23 Programmable I/O Lines
    - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
7. Operating Voltage = 1.8 – 5.5 V

8. Temperature Range = 40^0 to 85^0 C
9. Speed Grade = 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 - 5.5V
10. Power Consumption at 1MHz, 1.8V, 25^0 C
    - Active Mode: 0.2mA
    - Power Down Mode: 0.1 microA
    - Power-save Mode: 0.75 microA

2.4 ULTRASONIC SENSORS

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function. The basic principle of work:

1. Using IO trigger for at least 10us high level signal,
2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
3. IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

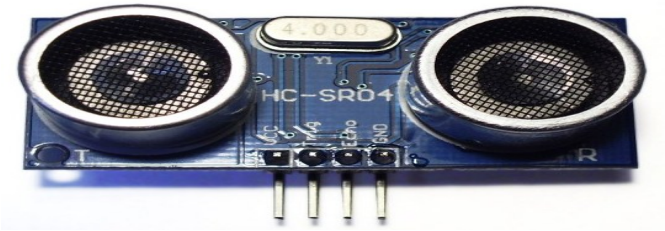


Figure 2.3 HC-SR04 Ultrasonic Sensor (Courtesy: www.micropic.com)

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

2.4.1 Electric Parameters:

Table 2.1 Electrical Parameters of HC-SR 04.

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

2.4.2 Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

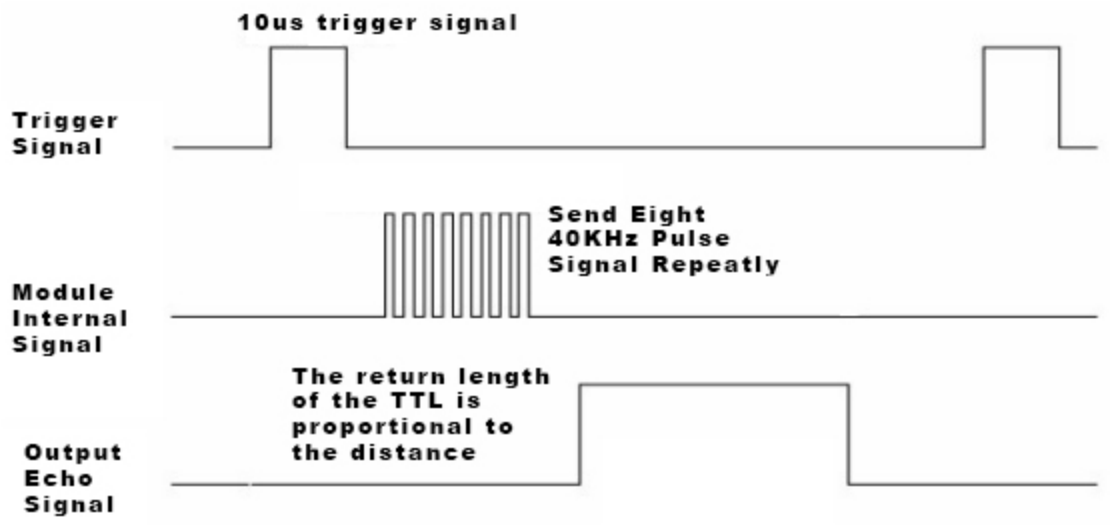


Figure 2.4 Timing Diagram of HC-SR 04(Courtesy: www.micropic.com)

2.5 ETHERNET CONTROLLER - ENC28J60

2.5.1 Ethernet controller features

1. IEEE 802.3 compatible Ethernet controller
2. Integrated MAC and 10BASE-T PHY
3. Receiver and collision squelch circuit
4. Supports one 10BASE-T port with automatic polarity detection and correction
5. Supports Full and Half-Duplex modes
6. Programmable automatic retransmit on collision
7. Programmable padding and CRC generation
8. Programmable automatic rejection of erroneous packets
9. SPI™ Interface with speeds up to 10 Mb/s

2.5.2 Operational

1. Two programmable LED outputs for LINK, TX, RX, collision and full/half-duplex status
2. Seven interrupt sources with two interrupt pins
3. 25MHz clock
4. Clock out pin with programmable prescaler
5. Operating voltage range of 3.14V to 3.45V
6. TTL level inputs
7. Temperature range: -40°C to +85°C Industrial, 0°C to +70°C Commercial (SSOP only)
8. 28-pin SPDIP, SSOP, SOIC, QFN packages.

2.5.3 Medium Access Controller (MAC) Features:

1. Supports Unicast, Multicast and Broadcast packets
2. Programmable receive packet filtering and wake-up host on logical AND or OR of the following:
   - Unicast destination address
   - Multicast address
   - Broadcast address
   - Magic Packet™
   - Group destination addresses as defined by 64-bit hash table
   - Programmable pattern matching of up to 64 bytes at user-defined offset
3. Loopback mode

2.5.4 Serial Peripheral Interface (SPI)

The ENC28J60 is designed to interface directly with the Serial Peripheral Interface (SPI) port available on many microcontrollers. The implementation used on this device supports SPI mode 0,0 only. In addition, the SPI port requires that SCK be at Idle in a low state; selectable clock polarity is not supported. Commands and data are sent to the device via the SI pin, with data being clocked in on the rising edge of SCK. Data is driven out by the ENC28J60 on the SO line, on the falling edge of SCK. The CS pin must be held low while any operation is performed and returned high when finished.

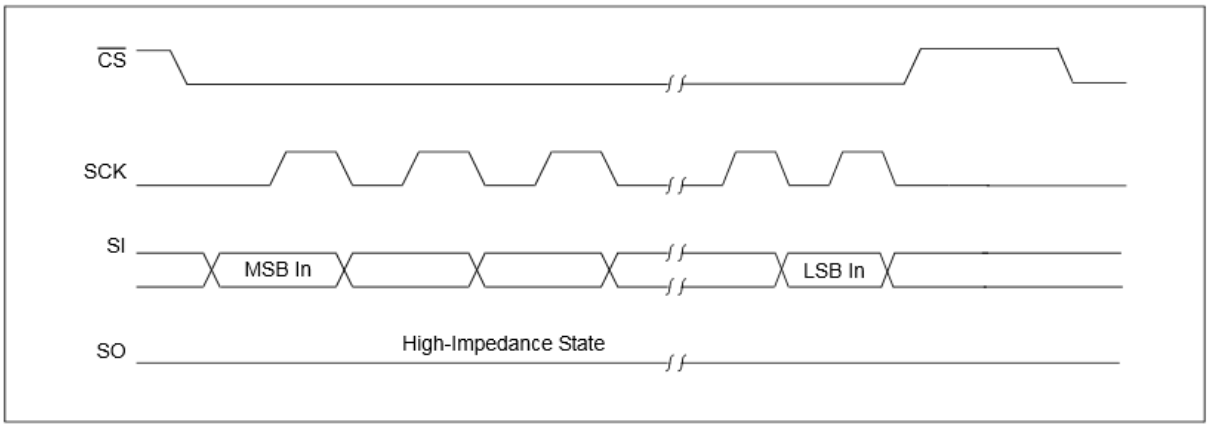


Figure 2.5 SPI Input Timing Diagram (Courtesy: www.microchip.com)

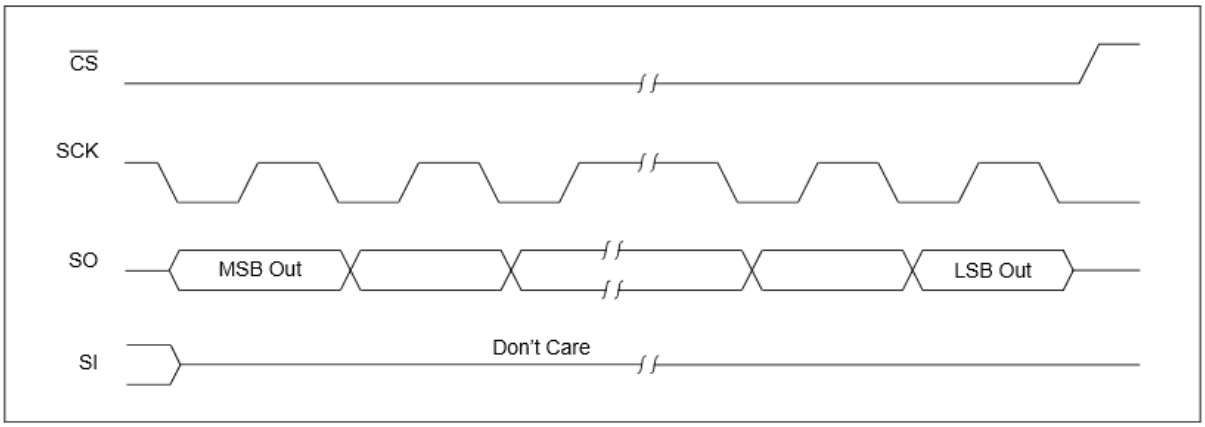


Figure 2.6 SPI Output Timing Diagram (Courtesy: www.microchip.com)

CHAPTER 3

DESIGN AND DRAWING

3. DESIGN AND DRAWING

3.1 INTRODUCTION

We have designed a 2-Dimensional Ultrasonic Anemometer. It has four ultrasonic sensors in pairs mounted opposite to each other and designated as North-South and East-West. Hence, we required four HC-SR 04 modules, modified to suit our requirements. Modification done was that the transmitter and receiver was de-soldered and placed opposite to each other. To connect to the internet, we require an Ethernet Controller (ENC28J60) and a modem. We connect them by RJ-45, category 6 cables.

3.2 BLOCK DIAGRAM

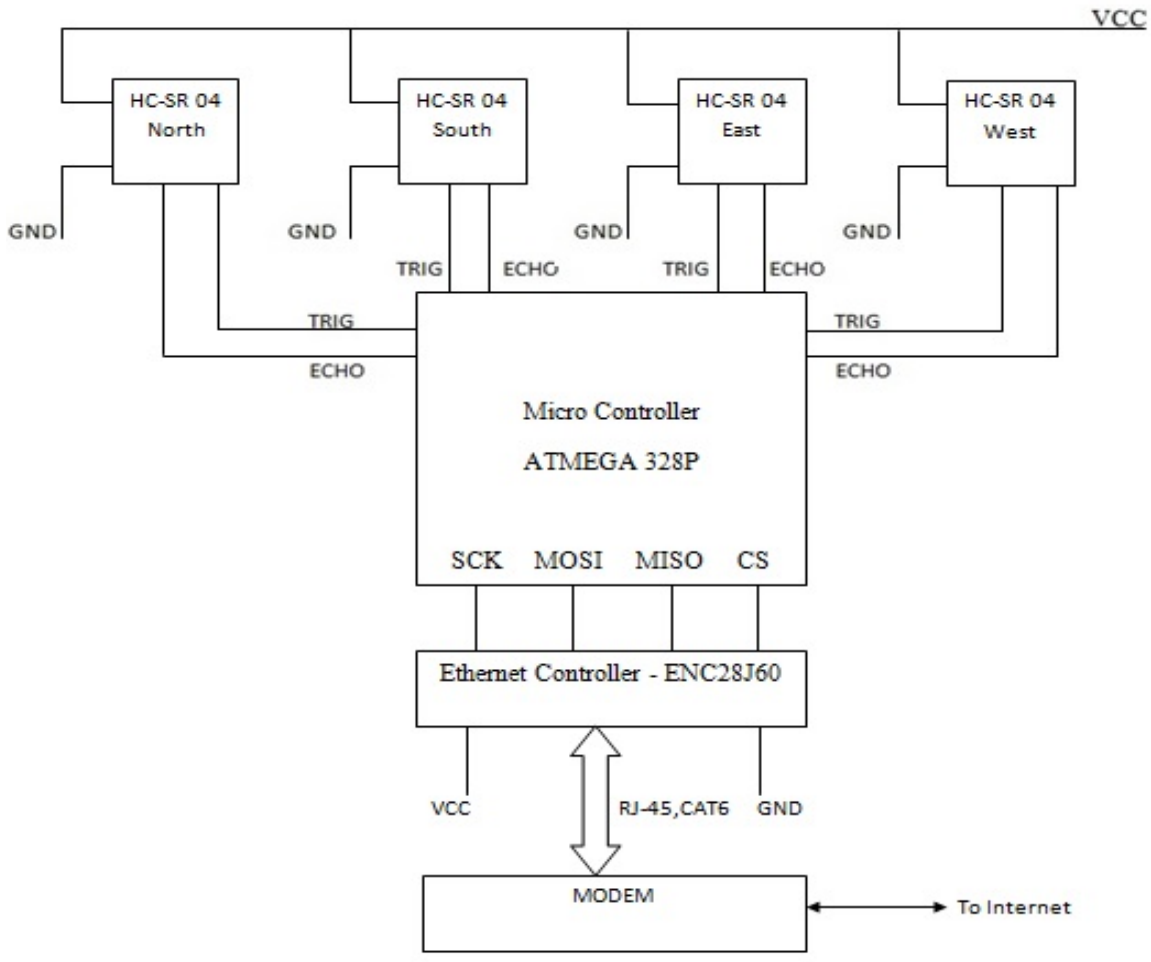


Figure 3.1 Block Diagram of the System

The ultrasonic sensor modules have four external pins – VCC, GND, TRIG and ECHO. VCC is connected to 5V supply and GND is connected to 0V. TRIG pin sends a 40KHz pulse towards receiver and ECHO pin sends a pulse when the pulse sent by the transmitter is received by the receiver. TRIG and ECHO pins are connected to the microcontroller, so that timing can be calculated. The Ethernet Controller is connected to the microcontroller via SPI. The SPI interface consists of four signals – SCK, MOSI, MISO and CS. These four signals are connected to the microcontroller.

3.3 PRINCIPLE

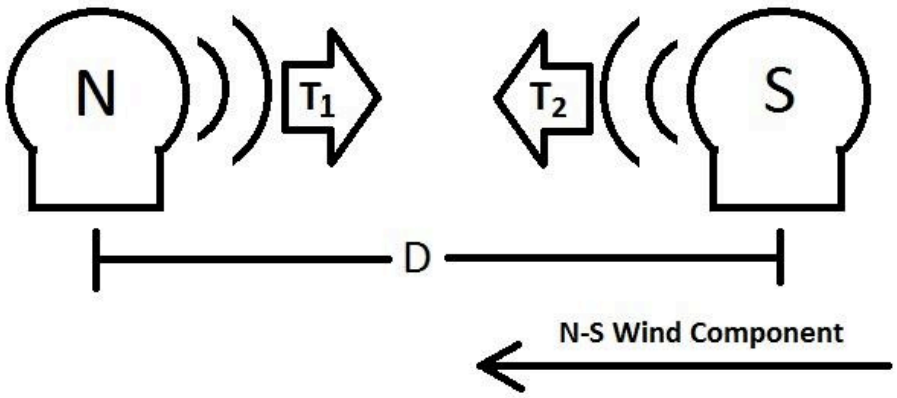


Figure 3.2 North-South Components and Wind Component

The sound wave of 40KHz has a constant velocity when there is no wind blowing. Consider North-South transceivers, where the transmitter emits a pulse towards the receiver. When the wind component is in the opposite direction of the transmitted pulse, it will oppose the sound wave. Thus, the velocity of the transmitted wave decreases and the time required to receive it increases. We can calculate the component of the wind blowing, since the distance between the opposite ultrasonic sensors is constant by simple calculations explained in the next

section. In the same way, we can calculate the other wind component and using vector mathematics, we can calculate the resultant wind speed as well as the wind direction.

3.3.1 CALCULATIONS

$T1 = D/(Vs+Wn)$ and $T2 = D/(Vs-Wn)$

Where,

Vs = Speed of Sound

Wn = Speed of North-South Wind Component

Therefore,

$Vs - Wn = D/T1$ ---------- (1)

$Vs + Wn = D/T2$ ---------- (2)

Subtracting (1) from (2),

$Wn = (D/2)*(1/T1 - 1/T2)$

Similarly, we can find East-West Wind Component - 'We'

Total Wind Speed => $W = (Wn^2 + We^2)^{0.5}$

Total Wind Direction => $\theta = \tan^{-1}(We/Wn)$

3.4 STRUCTURE – MECHANICAL DESIGN

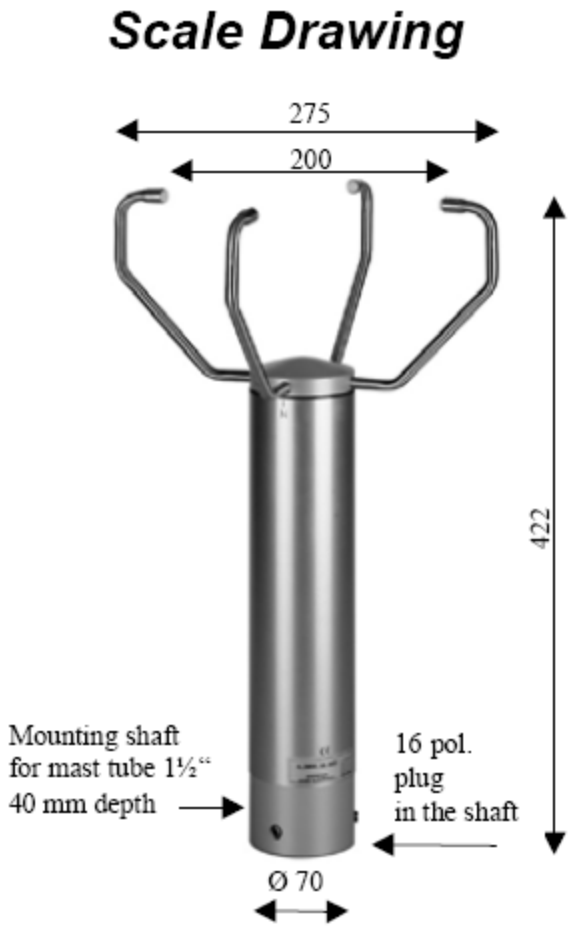


Figure 3.3 Ultrasonic Anemometer from Gill Instruments

The Indian Meteorological Department had given us this model as a reference point to begin our design and to explain us the concept of anemometer. The model is cylindrical in shape made out of Stainless Steel. The arms used to hold the sensors are also of hexagonal in shape. Here we see that the distance between the opposite transceivers is 200mm and the height of the device is 442mm. They have used an enclosure having a diameter of 70mm. However, in this design, the device is connected to the computer via an RS232 interface so as to log the readings into the computer. Our problem statement was to make this device accessible remotely. So, we decided to introduce a microcontroller and an Ethernet controller in the design so that the data can be uploaded to the internet hence, making the device remote access.

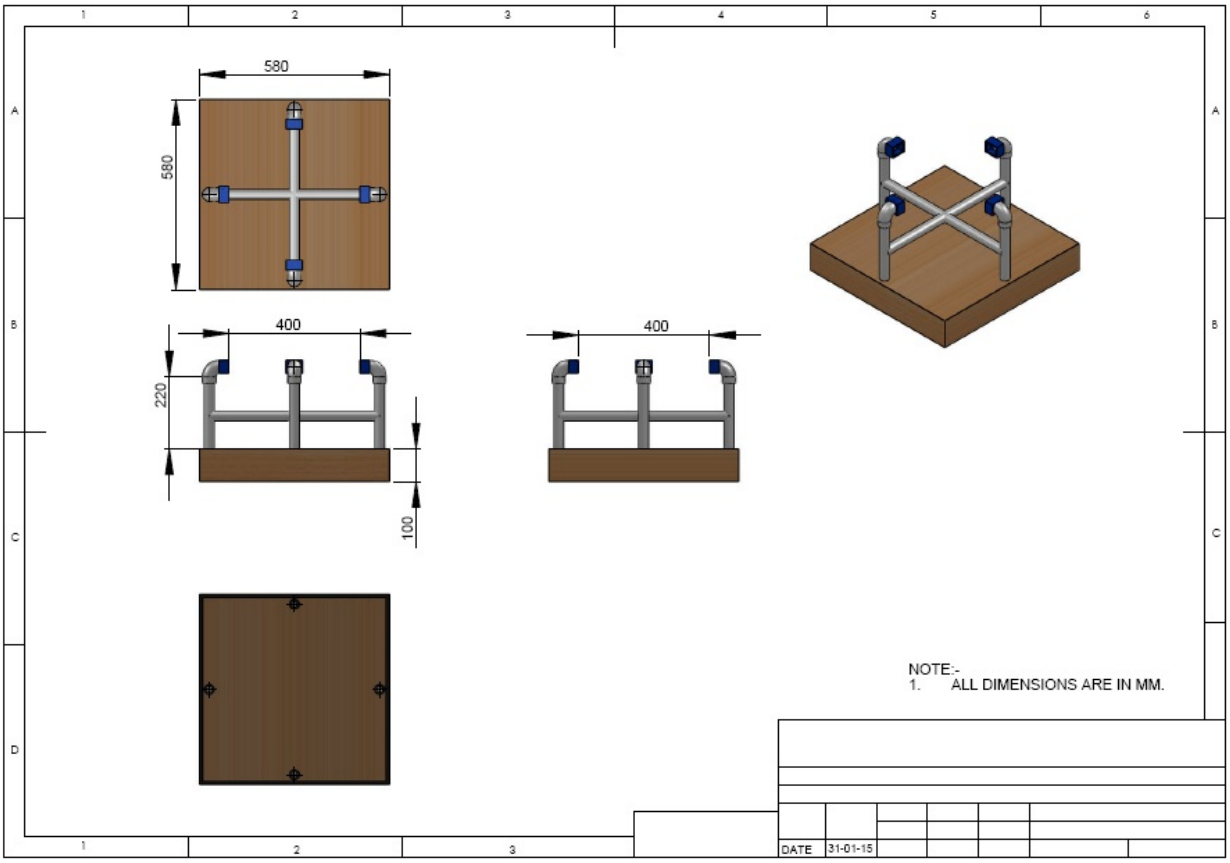


Figure 3.4 Initial Mechanical Design of the Structure

The requirements of the Indian Meteorological Department were to make the device remote access and that the entire system should be low cost and free of licensing and contracts. Keeping this in mind, we selected low cost sensors making a tradeoff between sensitivity and distance between sensors. The sensors that we used were low cost and hence having lower sensitivity than the ones used by Gill Instruments. The cost of such ultrasonic anemometers ranges between Rs 1,00,000/- to Rs 2,00,000/-. Our design costs around Rs 3,000/-.

Given in the above figure is the initial mechanical design that we made of our system. By measurements and observations, we found out that the sensors were most sensitive at the distance of 400mm. The experimentation carried out to find the sensitivity of sensors has been explained in the section ‘Experimentation’. Because of this, the dimensions of our device

became 580mm X 580mm. The distance between opposite transceivers is 400mm and the height of the device is 320mm. The height was decided arbitrarily.

The material used by Gill Instruments is SS (Stainless Steel) which is costly. We have used MS (Mild Steel) so as to reduce the cost of our device. SS is weather proof. We have painted our device with duco paint to make it weather proof and aesthetically appealing. We can also powder coat the device to fulfill this requirement. Cylindrical structure is aerodynamic but difficult to manufacture. We have made a rectangular design of our device so as to make its manufacturing easier and low cost. This reduces the cost of the device, without having an adverse effect on its operation in any way.

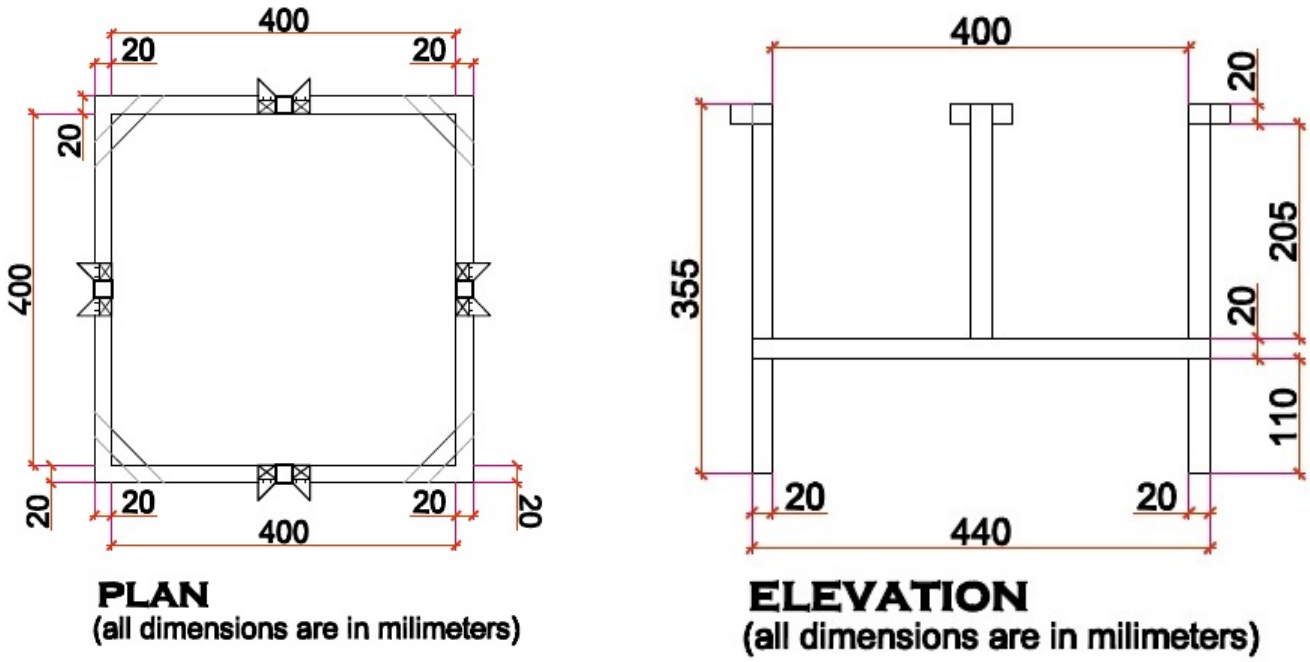


Figure 3.5 Final Mechanical Design of the Structure

The figure above shows the final design of our structure. The entire structure is square in shape and manufactured in SS. The pipes used as vertical arms are 205mm in length and have a square bore of 20mm X 20mm. The same pipes are used to make the square base frame having inner dimensions of 400mm X 400mm and outer dimensions of 440mm X 440mm. The height of the entire structure is 355mm. The sensors are fitted inside trapezoidal pieces made by cutting the same pipe. Trapezoidal shape is used for aesthetics. The parallel faces of the trapezoid are 20mm and 40mm respectively and non parallel faces have angles 90^0 and 45^0 respectively. The base frame is enclosed using the material Bakelite to protect the circuitry inside from weather changes. The legs of the structure have been fitted with leveling screws so as to adjust the device horizontally.

CHAPTER 4

MANUFACTURING

4. MANUFACTURING

4.1 INTRODUCTION

Since the structure was to be made out of Mild Steel, we had to fabricate it, cut it, drill it and weld it. We had decided to prepare the structure from MS pipes of 20mm thickness, because it was the size that could properly house our ultrasonic sensors. We decided to use Bakelite to enclose the structure to protect it from weather changes. We decided to make a drawer that would fit in the square frame to carry all our circuitry and battery, i.e. power supply.

4.2 PROCESS

4.2.1 FRAME

Figure 4.1 Frame of the Structure

We first procured the 20mm pipes as required by our design. We cut these pipes into pieces of 400mm for the frame, 110mm for the legs and 205mm for the arms that hold sensor wires. The ends of these pipes were cut at 45^0 and joined together by the process of spot welding. Thus, we manufactured a square frame of 400mm X 400mm dimensions. We welded the pipes of 205mm at the center of each side of the square as arms so as to hold the sensor wires. We also welded the legs of the frame which are basically pipes of 110mm length, as prescribed in the design.

Figure 4.2 Holes Drilled to pass Sensor Wires

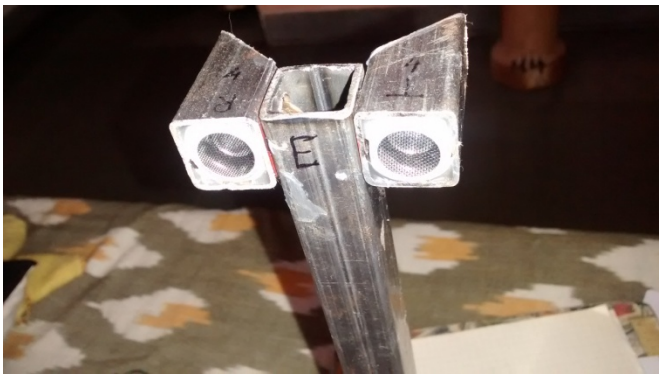


Figure 4.3 Trapezoidal Pieces holding the Sensors and wires passing through the Pipe

At the top end of the arms, we attached small trapezoidal pieces, manufactured from the same pipes so as to hold the ultrasonic sensors. We drilled holes in the frame wherever necessary so as to pass wires to the main circuitry inside the drawer.

4.2.2 SENSOR MOUNTING

The crucial step was to mount the sensors inside the trapezoidal pieces with perfection so that the readings will not get disturbed. They need to be aligned along the line of sight, perfectly and also checked for tilting vertically or horizontally. They should be exactly 400mm apart from each other. We used steel plates, protractors and spirit level to adjust the sensors exactly in place.

Figure 4.4 Silicone Sealant Cylinder

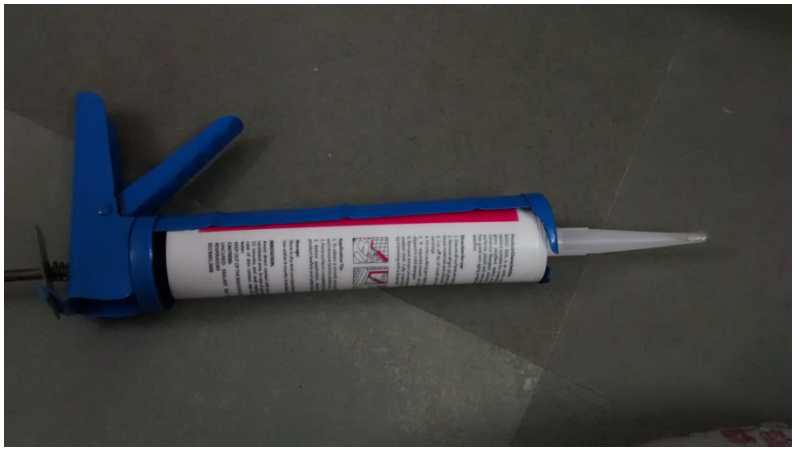


Figure 4.5 Silicone Sealant in its Gun

Figure 4.6 Silicone Sealant filled in the cavity to fit the sensors

We used the Silicone Sealant to hold the sensors in place inside their housing. Silicone sealant is an adhesive and is a very powerful, flexible product that can be used in many different applications. Silicone remains very flexible, even once it has fully dried or cured. This type of sealant can also withstand very high temperatures, making it ideal for applications that suffer high heat exposure. It is also rain proof, i.e. it will not allow water to pass and thus protecting our sensors from rain.

Figure 4.7 Wires passed through drilled holes

Each sensor has a pair of wires connecting it to the PCB. These wires have to be concealed to avoid wear and tear due to weather. The frame is drilled holes at appropriate locations and the wires are passed through these holes to the drawer containing the electronic circuitry.

4.2.3 Mother Board

On the development board, all the microcontroller pins are available for connections externally. We have manufactured a mother board, which is essentially a printed circuit board that has only those connections on it which are required for the operation, thus protecting all the other microcontroller pins. The required pins are: 2 digital I/O pins for each sensor circuit, 4 digital I/O pins for SPI interface between the microcontroller and Ethernet controller, and VCC-GND for all the components. We have had this mother board etched and all the components are mounted on it. There are also pins for providing external power supply.

CHAPTER 5

EXPERIMENTATION

5. EXPERIMENTATION

5.1 INTRODUCTION

The basic activity carried out during experimentation was to find out the optimum distance at which the sensors are most sensitive. Thus we carried out testing of the sensors at various distances and observed the output time period. The distance at which the output was most stable was noted as the distance of highest sensitivity.

5.2 SENSOR REMODELING

The HC-SR 04 Module is majorly used for distance measurement. So, in order to measure the time using the same module we had to remodel the sensor arrangements. Thus the sensors which were facing in the same direction were desoldered and positioned face-to-face along a line of sight. The sensors removed from the module were connected again to the module by parallel pair multi-thread wires. While re-soldering the sensors the polarity of the sensors were closely noted as mistake in this would be disastrous and can damage the whole module and make it useless. The wires of long length were taken so as to facilitate the accommodation of long distances comfortably. The sensors were placed away from the module and were connected to the module by wires.

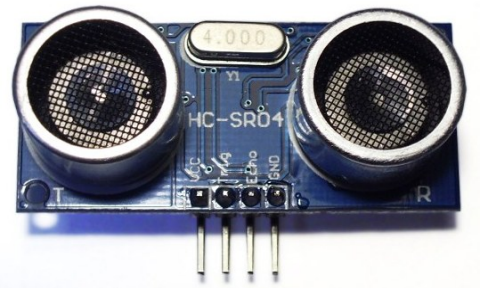


Figure 5.1 Original Structure of HC-SR 04 Module

Figure 5.2 Remodeled arrangement of HC-SR 04 module

5.3 SENSITIVITY OBSERVATION

We made a board so as to align the sensors on line of sight and to place them at different distances from each other.

We took a plywood board a scale on it and marked distances on the scale. We made arrangements so as to mount the sensors along the scale. Drawing the scale helped in two ways. It helped to mount the sensors in line of sight and keep a track of distance between the transmitter and receiver. Figures below show the board and the arrangement done to hold the sensors in place along the scale. The sensors (two pairs of Tx and Rx modules) were placed on a straight line path at different distances and tested rigorously. This test resulted that maximum sensitivity of sensors were observed at a distance of 400mm. Slightest variations in the wind were also observed, at a distance of 400mm.

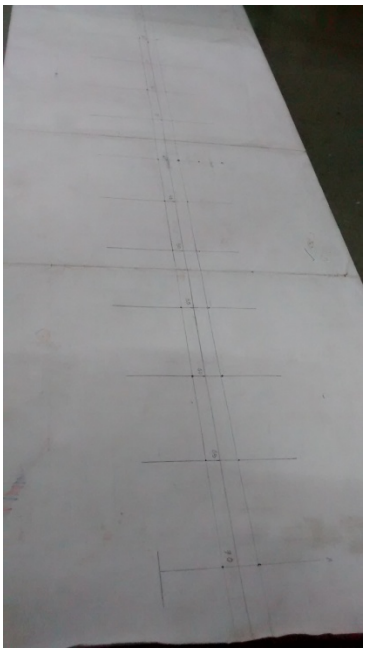


Figure 5.3 Scale marked for testing

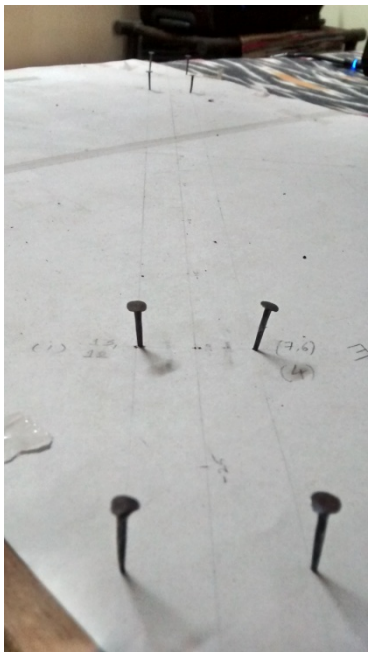


Figure 5.4 Arrangement done to mount the sensors on the board for testing

5.3.1 Observations

The sensors were connected to the microcontroller and placed firmly on the testing board. The program to measure the time period was burned in the microcontroller and the output was noted. The obtained output was then processed by Excel to represent the reading in the form of graph.

The graphs shown below represent the time period obtained at various distances between the sensors.

The graph below represents the readings at 200mm.

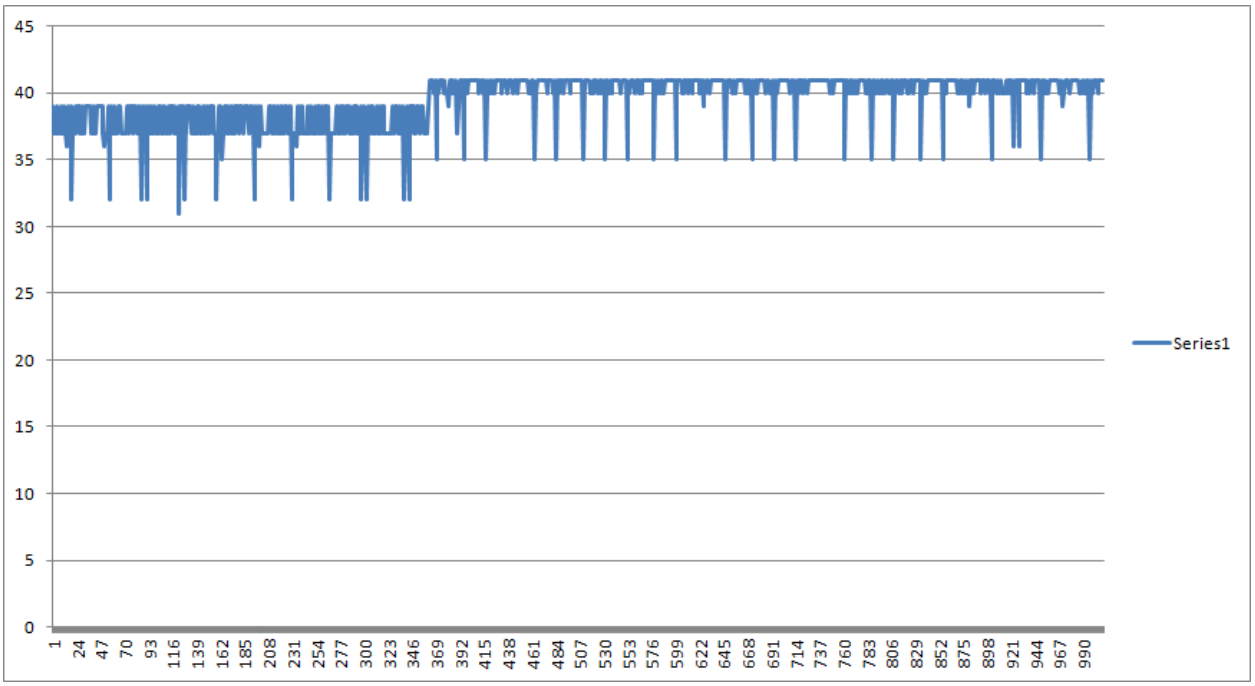


Figure 5.5 Readings at 200mm

The graph below represents the readings obtained at 400mm without disturbance

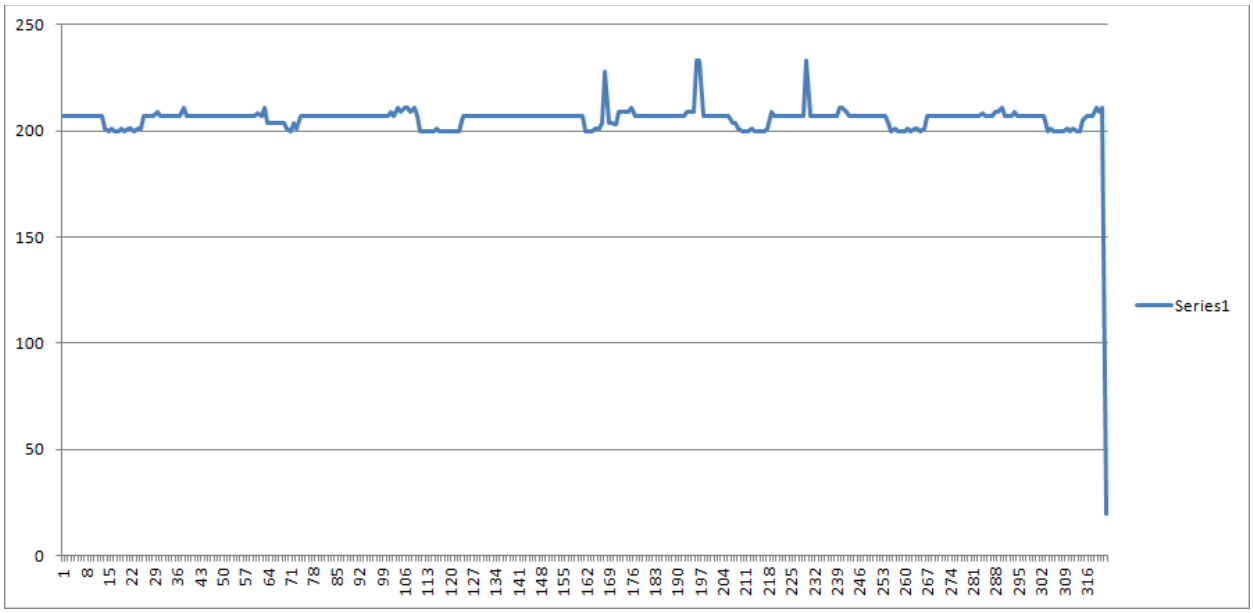


Figure 5.6 Readings at 400mm without disturbance

The graph below represents the readings obtained at 400mm with disturbance

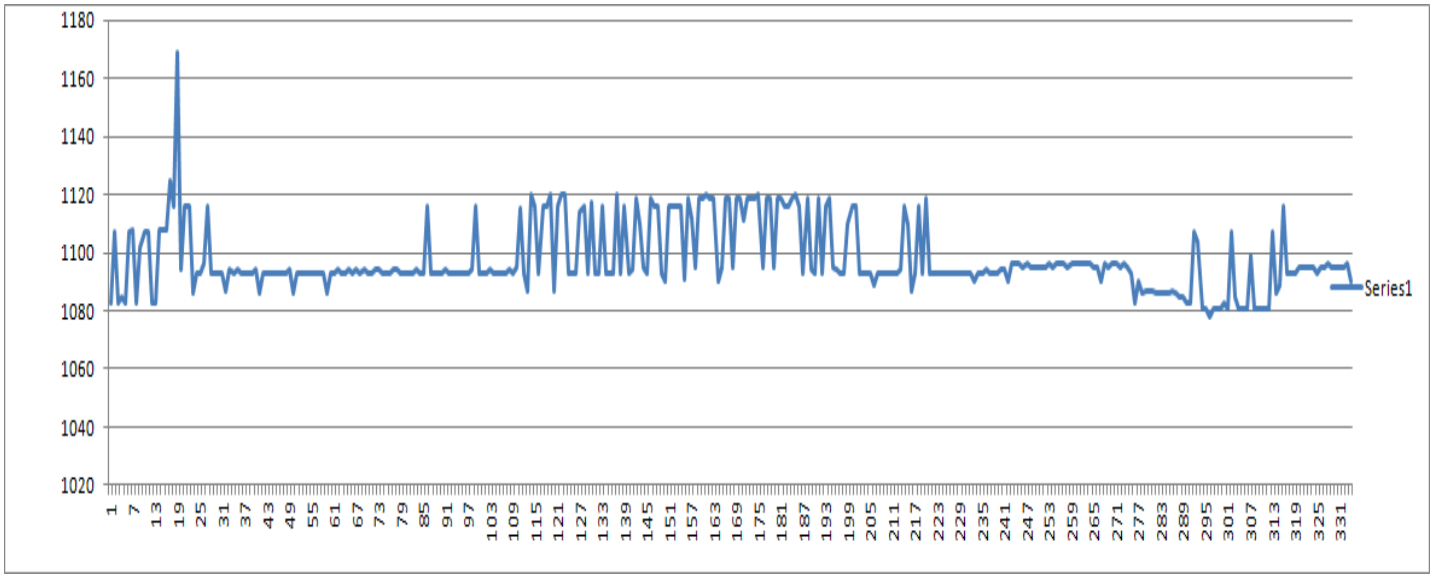


Figure 5.7 Readings at 400mm with disturbance

From the above graphs we can observe:

1. The reading obtained changes as the distance between the sensors changes.
2. The graph 1 represents the time period output obtained at 200mm. As observed the magnitude of the readings is small and changes rapidly even without any disturbance caused by wind. For the system to operate reliably the reading when no wind is flowing should be constant. If the readings are not constant then the microcontroller will not be able to distinguish between no wind and flowing wind. Thus distance between this range should be avoided for the Transmitter-Receiver pair.
3. Graph 2 represents the readings that are obtained when the distance between the sensors is 400mm. The readings obtained are quite stable over the range of time. Thus the microcontroller can distinguish between still wind and flowing wind. Though there are some spikes in the reading representing abrupt large changes in the reading, but these are very much less as compared to the reading in earlier conditions.
4. Graph 3 represents the readings obtained at 400mm when the disturbance is created by using a blower. These disturbances were introduced randomly and switched off after particular period of time. In the graph it is distinctly seen that the reading changes uniformly for a particular period of time and returns to normal after the disturbance. The change in the readings is distinct and thus the threshold points can be set to identify changes in the wind speed clearly.

Thus, from the above graphs we observe that the change in the reading can be obtained distinctly when the transmitter and receiver are placed 400mm apart from each other.

Thus in our design we have mounted the sensors at a distance of 400mm from each other.

5.3.2 Four Sensor Arrangement

To calculate the speed of the wind we need to resolve the vector component in to transverse components. To achieve this, the sensors should be arranged in perpendicular fashion so as to get the transverse components. The arrangement done can be seen in the figure below.

Figure 5.8 Four Sensor Test Board

Till now the observations obtained were single-sided i.e. by using only single pair of transmitter and receiver. Now we have to obtain the observations for 4 pairs of transmitter and receivers. While arranging these pairs care needs to be taken as to place the correct pair of transmitter-receiver on one side of the arrangement. This needs to be done so as to get correct reading form the microcontroller. The sensors we held firmly to their position with the help of nails. When the whole circuit is operational then we get reading on the terminal window. Since we have already found out the distance at which we get stable reading we need to observe all the values of the components. Once we get stable output values we need to generate a uniform disturbance in air so as to monitor the changes in the reading. The output values obtained when processed by using the Excel generate graph which is represented in figure below.

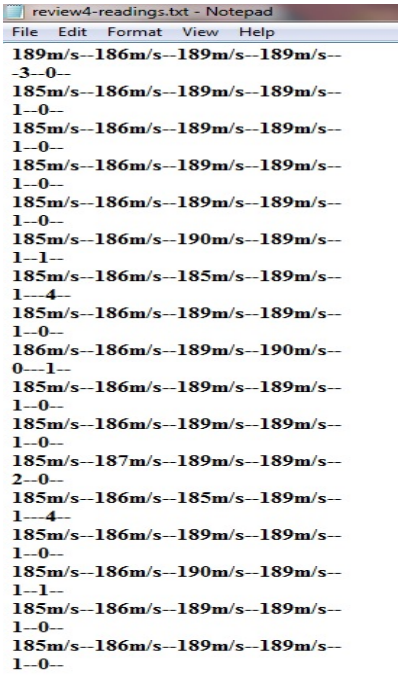


Figure 5.9 Readings obtained from four sensor arrangement

5.4. Communication Module

The communication module is the heart of the whole device as the readings obtained will be of no use if not processed by qualified operator or software. Both of these two things are not going to be present as the whole device is to be placed at remote locations. Thus to process the readings we need to send these to the operator via communication channel.

The communication channel should be reliable, robust and cheap to operate. Thus the communication channels shortlisted are:

1.Ethernet
2.GSM/GPRS

5.4.1 Ethernet

Figure 5.10 Ethernet Controller ENC28J60 (courtesy: www.microchip.com)

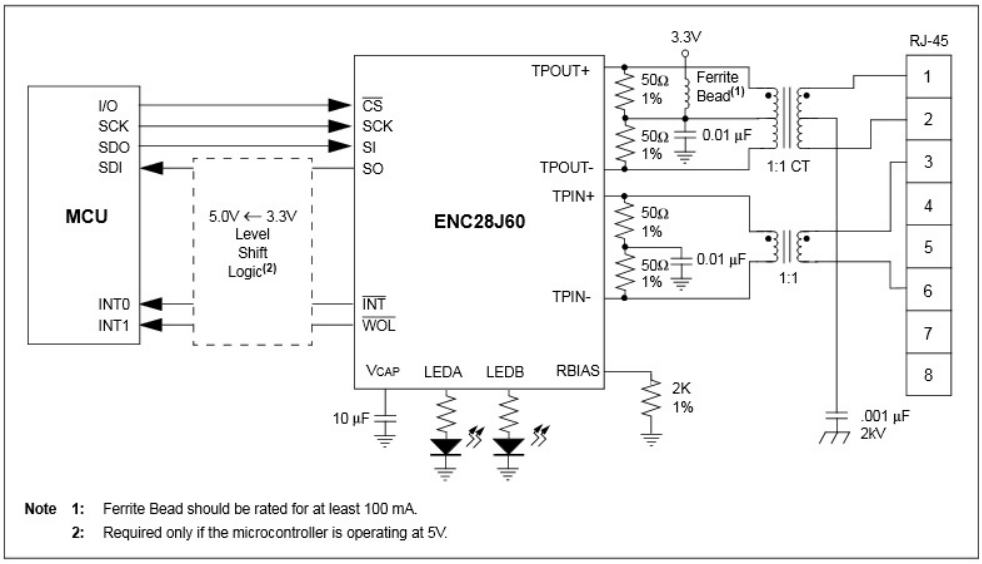


Figure 5.11 ENC28J60 Interface with MCU and RJ45 (courtesy: www.microchip.com)

The Ethernet connection that we have used here is an Ethernet shield. The data from the module is uploaded in a shared network. The Ethernet Shield is connected to the LAN along with the terminal that is used to burn the code in the microcontroller to a router. On burning the (for the IP generation) code we receive an IP address for the shield. The Shield has a permanent unique MAC address which is allocated to the shield by the manufacturer. Now we have an IP address as well as MAC address which completes the requirement for joining the Internet.

Since we are now connected on internet the next work of the microcontroller is to send the data to the respective IP address. In the main code the IP address generated earlier is mentioned as the IP address changes due to dynamic allocation of the IP address. So the IP address in the main code needs to be changed every time we disconnect from the Internet.

Once the data from the sensors is acquired and data is sent to the mentioned IP address, we need to observe the readings on a browser. For that open any internet browser (Google Chrome, Mozilla Firefox, Internet Explorer, etc) and search for the IP address. On successful connection we can see the content in HTML form. It refreshes after a particular interval of time as specified in the code to display new readings.

The advantages of using Ethernet are:

1. Reliable
2. Cost effective
3. Easy maintenance of the network

The disadvantages of Ethernet are:

1. Wired connection is necessary.
2. To connect to a remote device we need to place Ethernet cables.

5.4.2 GSM/GPRS

Figure 5.12 GSM/GPRS Module using SIM900 (courtesy: www.electrofreaks.com)

The other way of communicating with the control terminal is by a SMS using a CDMA/GSM transmitter-receiver i.e. a GSM module.

The GSM module works simply by using the mobile networks. To get a connection we need to provide a SIM card of a telephone operator so as to get a network. The controlling of the GSM module is done by AT commands or Hayes command.

The command set consists of a series of short text strings which combine together to produce complete commands for operations such as dialing, hanging up, and changing the parameters of the connection. The vast majority of dialup modems use the Hayes command set in numerous variations.

By using the AT commands the modem itself could switch itself between one of two modes:

1. Data mode in which the modem sends the data to the remote modem. (A modem in data mode treats everything it receives from the computer as data and sends it across the mobile network).

2. Command mode in which data is interpreted as commands to the local modem (commands that the local modem should execute).

CHAPTER 6

RESULTS AND DISCUSSION

6. RESULTS AND DISCUSSION

6.1 INTRODUCTION

We carried out the experimentation as described in the previous chapter and obtained the results which are summarized in this chapter. We also have analyzed the results and the relevant discussions have been given hereafter.

6.2 RESULTS

After rigorous testing and implementation of our design the following are the results obtained:

- HC-SR 04 low-cost ultrasonic sensor, primarily used for distance measurement are modified and implemented for measuring the time required for ultrasonic ray to travel a specified distance. Instead of placing the sensors in the same direction they are modified in such a way that Tx-Rx pair of a module is placed in front of each other.
- These low-cost sensors operate accurately at a distance of 400mm. So a chassis framework of MS (Mild Steel) with pillars erected at a distance of 400mm is constructed. The frame work is balanced using leveling screws foe non-horizontal surfaces.

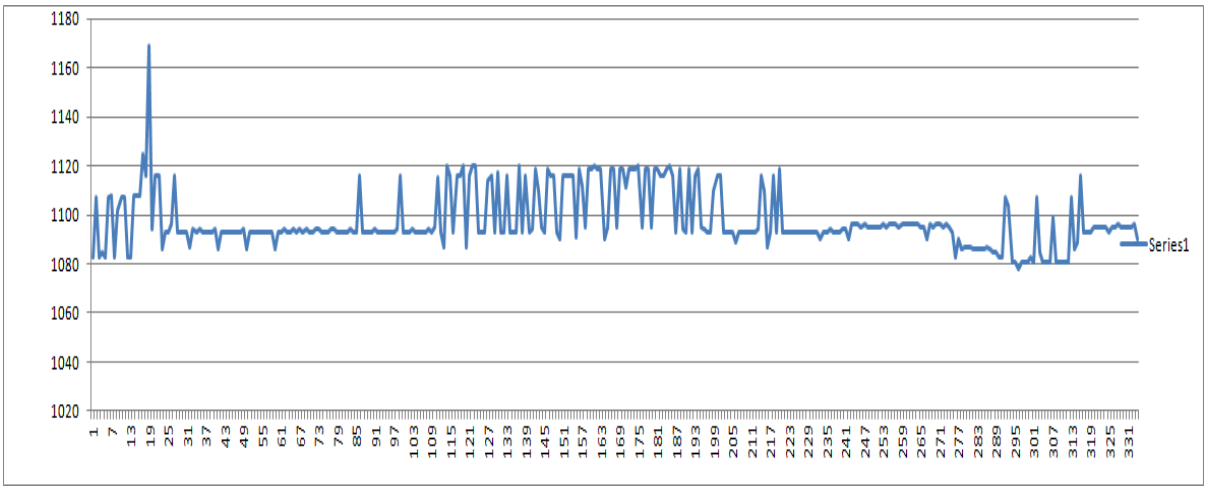


Figure 6.1 Results of sensor testing at 400mm with disturbance of wind

- When all the sensors were positioned and aligned at 400mm, the above graph was obtained. It shows a measurable deflections in the readings when wind flows through.
- The following are the readings obtained when all the four sensors are activated for calculating the overall wind speed and direction.

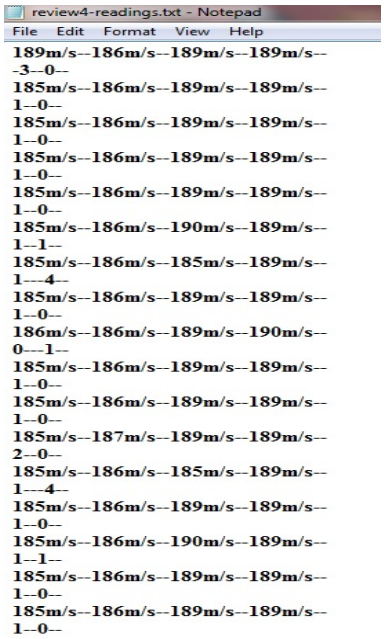


Figure 6.2 Readings obtained with all sensors activated

The readings in m/s are the four wind speeds calculated in North-South, South-North, East-West and West – East directions respectively. The two numbers below them show the X-component of wind (in North-South direction) and Y-component (in East-West direction) respectively. These components are obtained by subtracting wind speeds in the same direction. If no wind is blowing, these X and Y components show ‘0’ value and they get deflected with the presence of wind.

6.3 DISCUSSION

- The square void area between the sensor pillars is majorly used for placing the entire electronics and the heart of the entire instrument. The entire electronic circuitry is placed in a drawer like structure which can be removed for servicing the instruments; as well the square void framework protects the circuitry for mechanical shocks and damages.
- The entire structure is coated with a duco-paint in order to avoid rusting of the instrument.
- The entire instrument is powered using a Li-Po battery, giving output voltage of 5V and a current output of 1.5Amp. These batteries are portable batteries, as our instrument is a remote one, such portable batteries are best for use.
- A communication module is necessary to transfer the data collected by the ultrasonic anemometer from the site to the control room in the meteorological department.
- Earlier a GSM Module which used SIM900 was employed in the system. But, this module proved to be incompatible in areas where there was no data connection. Also, the speed of data transfer was also less and insufficient to our needs.
- So, a reliable wireless transfer of data was not possible using GSM module or SIM900.
- Another way of communicating via wireless media was by using buoys communication module which is majorly used in Tsunami detection instruments. As, our sponsors were the Indian Meteorological Dept. we consulted them with this idea, but they discouraged it. As, their domain did not cover this field.
- Finally we thought of using a wired communication for transfer of data, as using a wired connection would be cheap, effective, and damage proof.
- Wired connections also enabled us a faster data transfer and a wider range of coverage.
- Wired connections also enabled us to use the INTERNET, which enabled us to transmit the data on a site address which was working in real-time.
- An Ethernet controller enabled us to put the data over internet using a wired network which was used to access the internet.
- Ultimately a transfer of data over a wired network proved reliable.
- This data was transferred over the internet using a dynamic site or a website which is updated in real-time.

- This site is updated after every 5 seconds.
- This website made it possible for the controller to access the reading of the instrument anywhere using the internet.

6.4 FUTURE ADVANCEMENTS

- The website can be used for further expansions, like when one or more anemometers are installed, these instruments can be connected to each other in a network and the entire weather forecast of the region can be determined.
- This enables weather reporting not just too the national weather stations, but also to the local weather stations.
- Also this information or data can be made available to the common people, as it is on the internet.
- One more application of this instrument is that it is used in the aviation industry.
- During, take-off and landing the air turbulence on the runway is very important for the pilot to maneuver the plane for safe take-off and landing.
- So, anemometers installed at the start and at the end of the runway are useful for determining the air turbulence at the ground.
- The data collected from these instruments can be compiled and then sent to the control room and then finally conveyed to the pilot who is supposed to land or fly the aircraft safely.
- So, a fairly accurate pattern of wind turbulence is necessary for the safety of pilot.

CHAPTER 7

CONCLUSION

7. CONCLUSION

The problem statement given to us by the Indian Meteorological Department was to build an ultrasonic anemometer so that the data can be logged to a remote location so that there is no requirement of a wired connection between the device and computer.

We built the anemometer using the ultrasonic sensors HC-SR04 by remodeling them as per our requirements to make the transmitter and receiver face each other. We used the ATMEGA328p to control the operations of the four sensors and get readings from them in microseconds.

We measured the sensitivity of the sensors by trial and error method by mounting them on the measurement board at different distances from each other and observing the stability of readings and deflection of readings in presence of wind. We found that the sensors were most sensitive at a distance of 400mm from each other.

We now manufactured the structure which is a square frame having dimensions 400mm X 400mm. We mounted the sensors on four arms using silicone sealant. The entire structure was manufactured in Mild Steel (MS) and painted using duco-paint so as to protect it from weather changes. The drawer containing all the electronic circuitry was manufactured that fits into the square void in the frame. We installed a magnetic compass on the frame to orient the device perfectly and a spirit level to level the device horizontally.

We then implemented the GPRS (SIM900) to transfer the data using sim card data to a remote location via internet. However, this data transfer was found to be unreliable, since the wireless data network fluctuated rapidly. This made the data to not get transferred at all when there was no network coverage.

We then shifted to Ethernet controller, which connects to the microcontroller via SPI (Serial Peripheral Interface) and to the modem via RJ45 Category 6 cable. This enables us to access the internet and log the data to a remote location via the internet. This can enable us to add security by restricting the access to limited personnel only.

REFERENCES

1. Wind Observer II (Gill Instruments) Datasheet

2. Wind Observer 65 (Gill Instruments) User Manual

3. “Physical and logistical considerations of using ultrasonic anemometers in aeolian sediment transport research” - Ian J. Walker

4. “To Measure Wind Speed using the theory of One-dimensional Ultrasonic Anemometer” – Zhou Yufeng, Wang Yan

5. www.atmel .com (ATMEGA 328p - Datasheet)

6. www.technik.dhbw-ravensburg.de

7. www.microchip.com (ENC28J6 – Ethernet Controller datasheet)

8. www.micropic.com (HC-SR04 Ultrasonic Sensor Datasheet)

9. www.wikipedia.org

APPENDIX A

BILL OF MATERIALS

Bill of Materials

Sr. No.	Component	Rate (Rs.)	Quantity	Cost (Rs.)
1.	Microcontroller - ATMEGA 328p	1200	1	1200
2.	Ultrasonic Sensor - HC-SR04	200	4	800
3.	Ethernet Controller - ENC28J6	300	1	300
4.	Mother Board PCB etching	200	1	200
5.	Power Supply	350	1	350
6.	Structure	500	1	500
7.	Silicone Sealant	100	1	100
8.	Duco-paint	200	1	200
	Total			3650

APPENDIX B

DATASHEETS

Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash

ATmega48A; ATmega48PA; ATmega88A; ATmega88PA; ATmega168A; ATmega168PA; ATmega328; ATmega328P

Features

- High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1KBytes EEPROM
  - 512/1K/1K/2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix® acquisition
  - Up to 64 sense channels
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA
  - Power-down Mode: 0.1µA
  - Power-save Mode: 0.75µA (Including 32kHz RTC)

1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

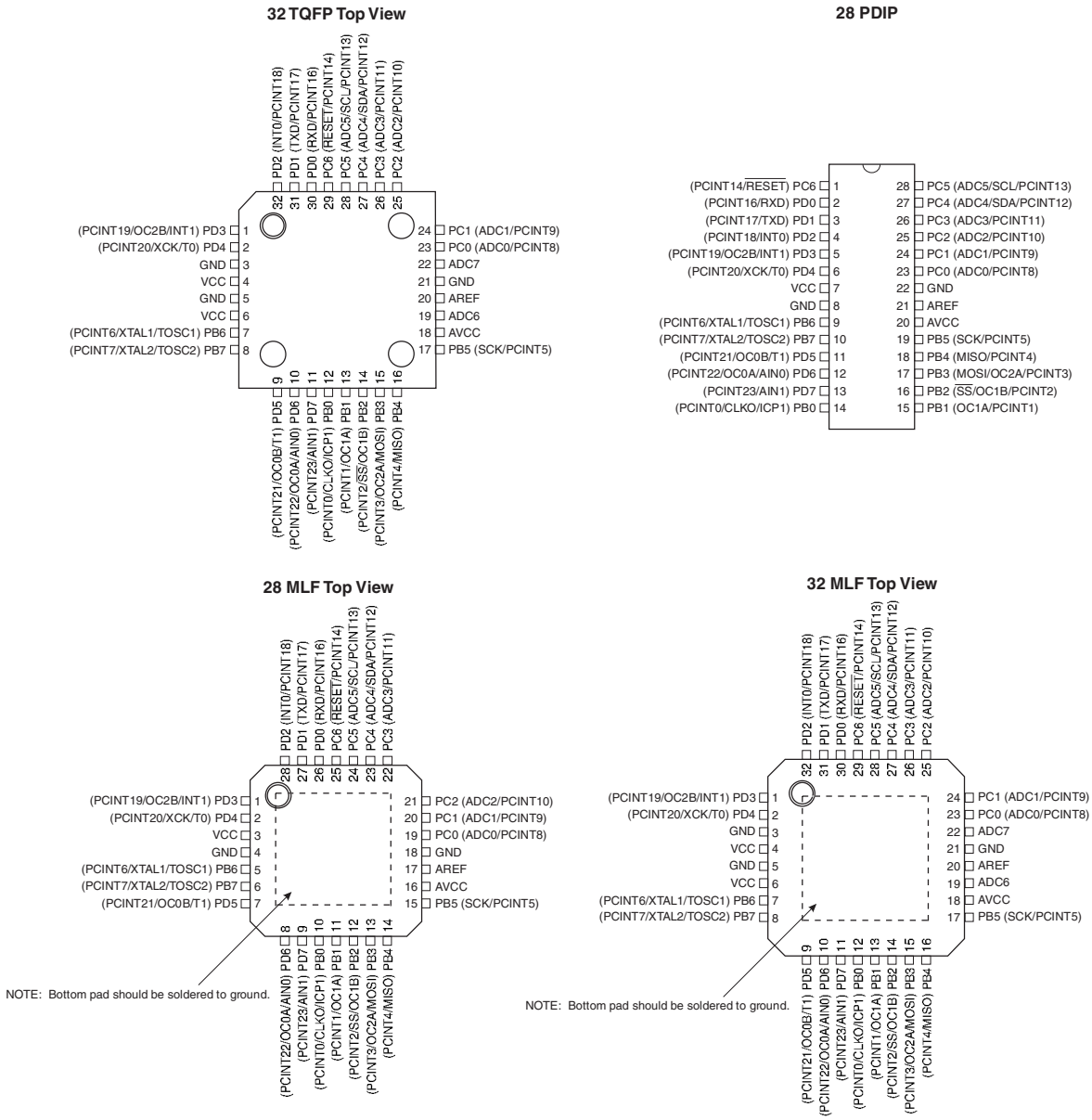


Table 1-1. 32UFBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ”Alternate Functions of Port B” on page 83 and ”System Clock and Clock Options” on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 29-16 on page 312. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ”Alternate Functions of Port C” on page 86.|

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in ”Alternate Functions of Port D” on page 89.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6...4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

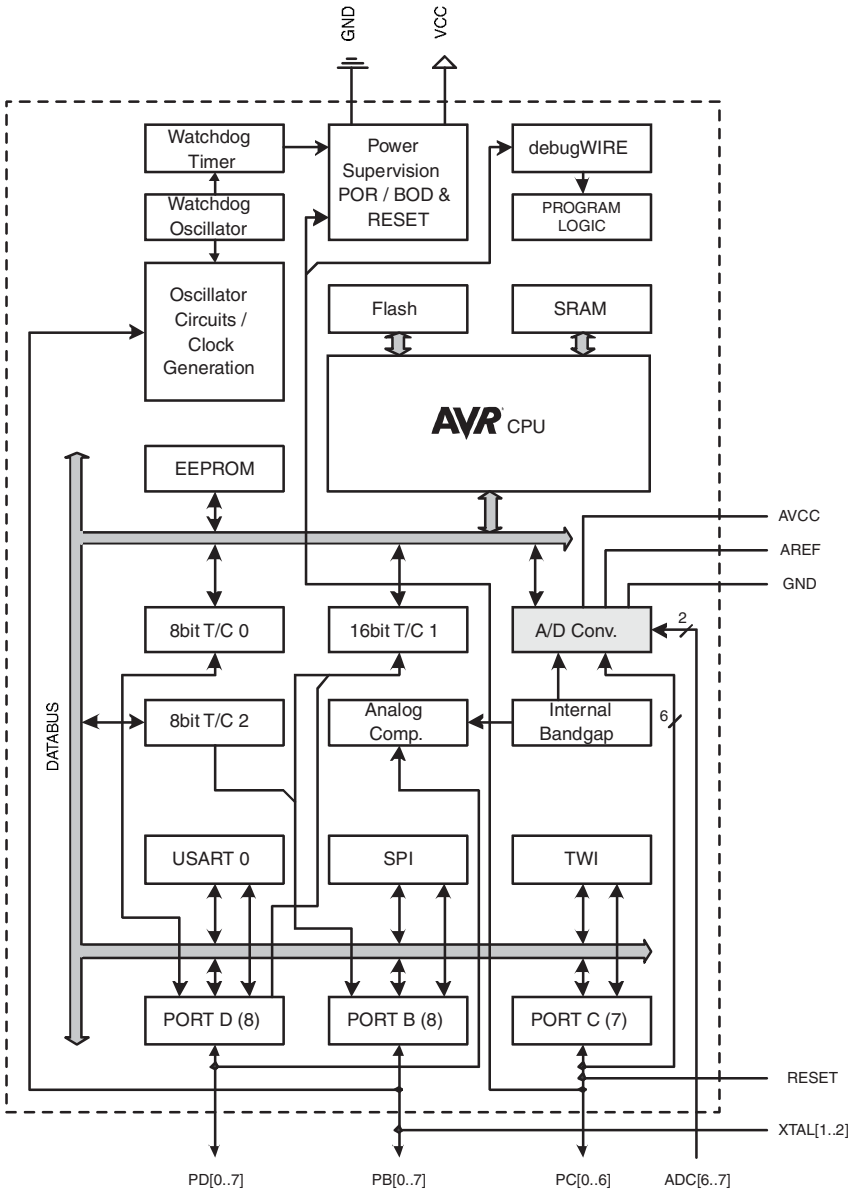
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48A/PA/88A/PA/168A/PA/328/P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48A/PA/88A/PA/168A/PA/328/P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



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 ATmega48A/PA/88A/PA/168A/PA/328/P provides the following features: 4K/8Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1Kbytes EEPROM, 512/1K/1K/2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, 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 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.

Atmel® offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR® microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile 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 ATmega48A/PA/88A/PA/168A/PA/328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48A/PA/88A/PA/168A/PA/328/P 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.

2.2 Comparison Between Processors

The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

ATmega48A/PA/88A/PA/168A/PA/328/P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega 48A/48PA there

is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

6. Capacitive Touch Sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the Atmel QTouch and Atmel QMatrix® acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API’s to retrieve the channel information and determine the touch sensor states.

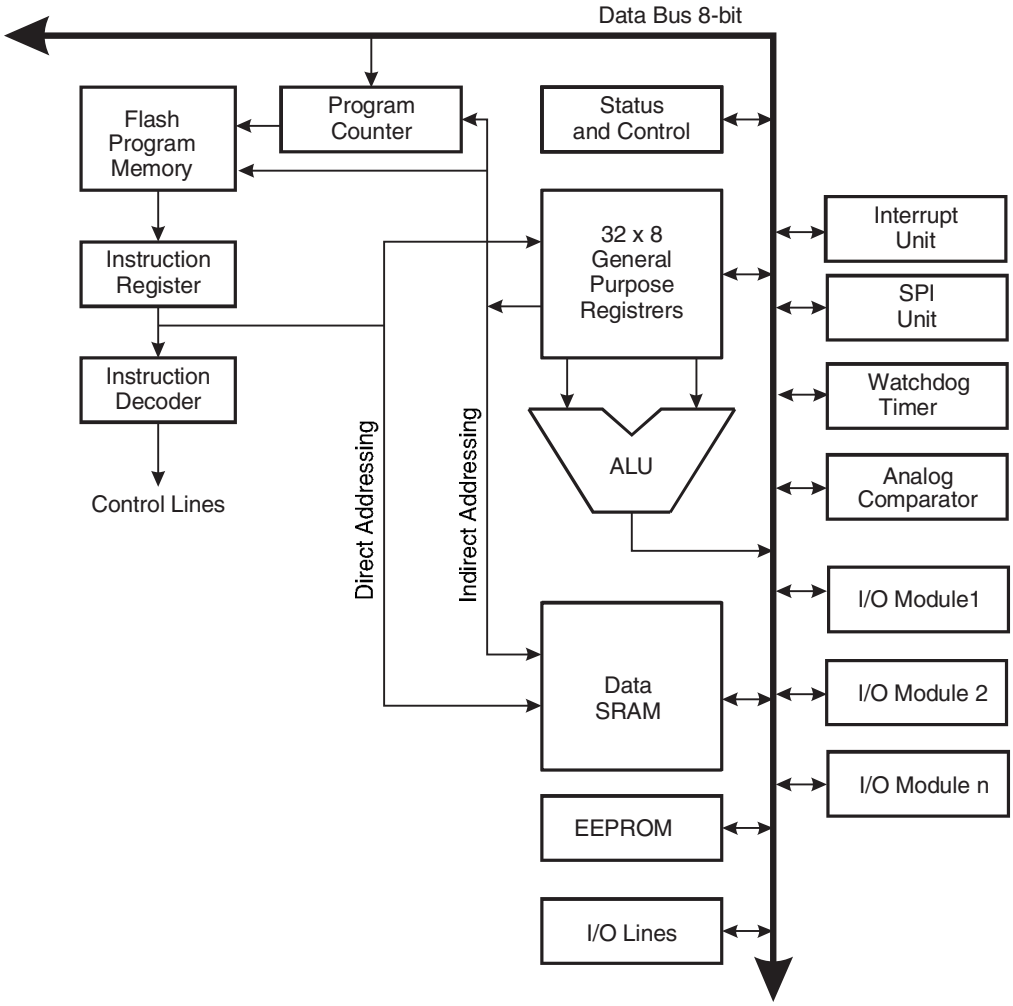
The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the Atmel QTouch Library User Guide - also available for download from Atmel website.

7. AVR CPU Core

7.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 7-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two oper-

ands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega48A/PA/88A/PA/168A/PA/328/P has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

7.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

7.3 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

15. 8-bit Timer/Counter0 with PWM

15.1 Features

- **Two Independent Output Compare Units**
- **Double Buffered Output Compare Registers**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch Free, Phase Correct Pulse Width Modulator (PWM)**
- **Variable PWM Period**
- **Frequency Generator**
- **Three Independent Interrupt Sources (TOV0, OCF0A, and OCF0B)**

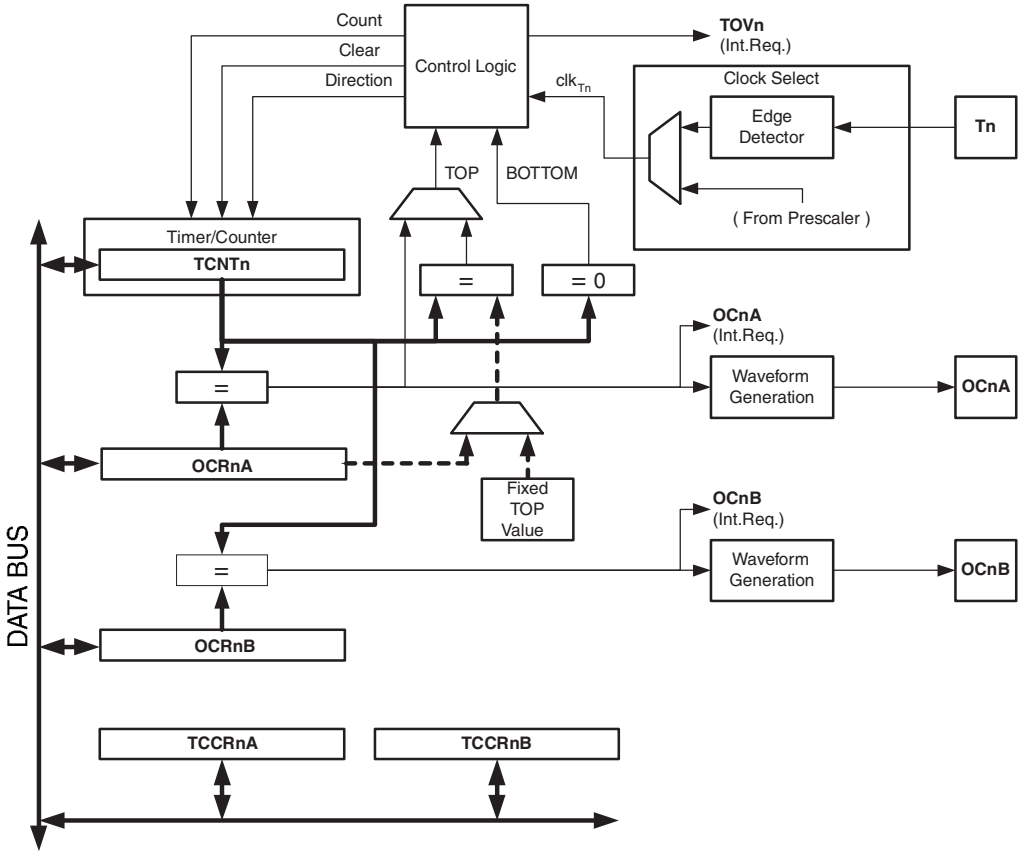
15.2 Overview

Timer/Counter0 is a general purpose 8-bit Timer/Counter module, with two independent Output Compare Units, and with PWM support. It allows accurate program execution timing (event management) and wave generation.

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 15-1. For the actual placement of I/O pins, refer to "Pinout ATmega48A/PA/88A/PA/168A/PA/328/P" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "Register Description" on page 105.

The PRTIM0 bit in "Minimizing Power Consumption" on page 41 must be written to zero to enable Timer/Counter0 module.

Figure 15-1. 8-bit Timer/Counter Block Diagram



15.2.1 Definitions

Many register and bit references in this section are written in general form. A lower case “n” replaces the Timer/Counter number, in this case 0. A lower case “x” replaces the Output Compare Unit, in this case Compare Unit A or Compare Unit B. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 15-1 are also used extensively throughout the document.

Table 15-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A Register. The assignment is dependent on the mode of operation.

15.2.2 Registers

The Timer/Counter (TCNT0) and Output Compare Registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Registers (OCR0A and OCR0B) are compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins (OC0A and OC0B). See ”Using the Output Compare Unit” on page 122 for details. The compare match event will also set the Compare Flag (OCF0A or OCF0B) which can be used to generate an Output Compare interrupt request.

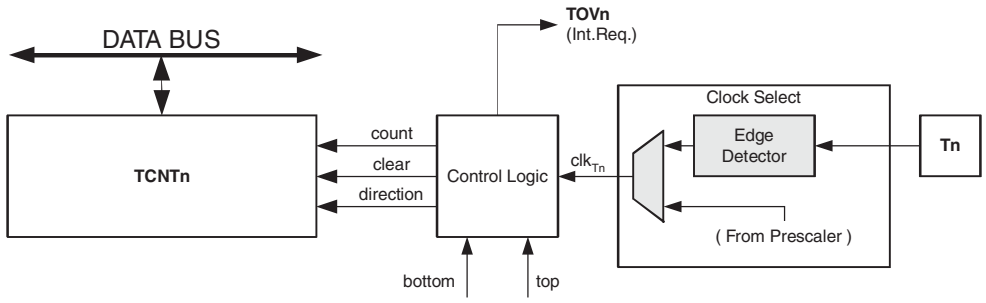
15.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0B). For details on clock sources and prescaler, see ”Timer/Counter0 and Timer/Counter1 Prescalers” on page 139.

15.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 15-2 shows a block diagram of the counter and its surroundings.

Figure 15-2. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
direction	Select between increment and decrement.
clear	Clear TCNT0 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T0} in the following.
top	Signalize that TCNT0 has reached maximum value.
bottom	Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0A) and the WGM02 bit located in the Timer/Counter Control Register B (TCCR0B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC0A and OC0B. For more details about advanced counting sequences and waveform generation, see ”Modes of Operation” on page 99.

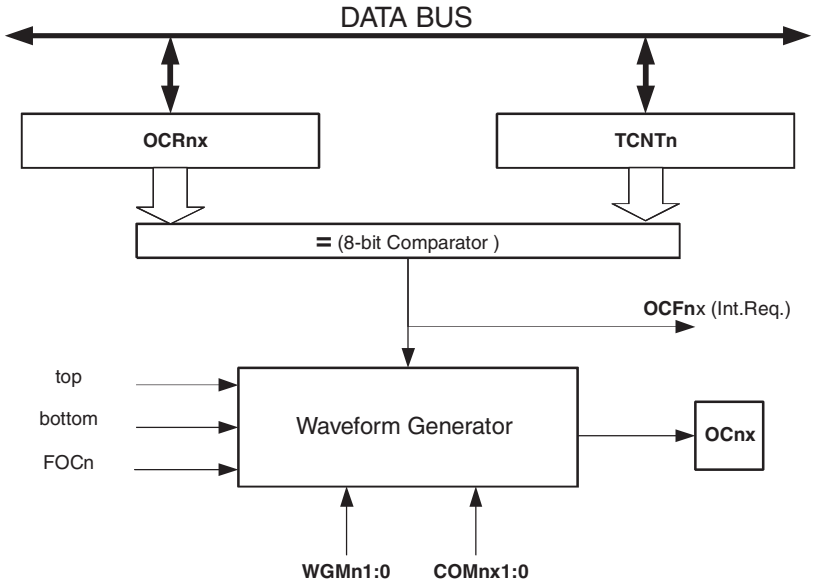
The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM02:0 bits. TOV0 can be used for generating a CPU interrupt.

15.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Registers (OCR0A and OCR0B). Whenever TCNT0 equals OCR0A or OCR0B, the comparator signals a match. A match will set the Output Compare Flag (OCF0A or OCF0B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM02:0 bits and Compare Output mode (COM0x1:0) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (”Modes of Operation” on page 99).

Figure 15-3 shows a block diagram of the Output Compare unit.

Figure 15-3. Output Compare Unit, Block Diagram



The OCR0x Registers are double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0x Compare Registers to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0x Buffer Register, and if double buffering is disabled the CPU will access the OCR0x directly.

15.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0x) bit. Forcing compare match will not set the OCF0x Flag or reload/clear the timer, but the OC0x pin will be updated as if a real compare match had occurred (the COM0x1:0 bits settings define whether the OC0x pin is set, cleared or toggled).

15.5.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0x to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

15.5.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the Output Compare Unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0x value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

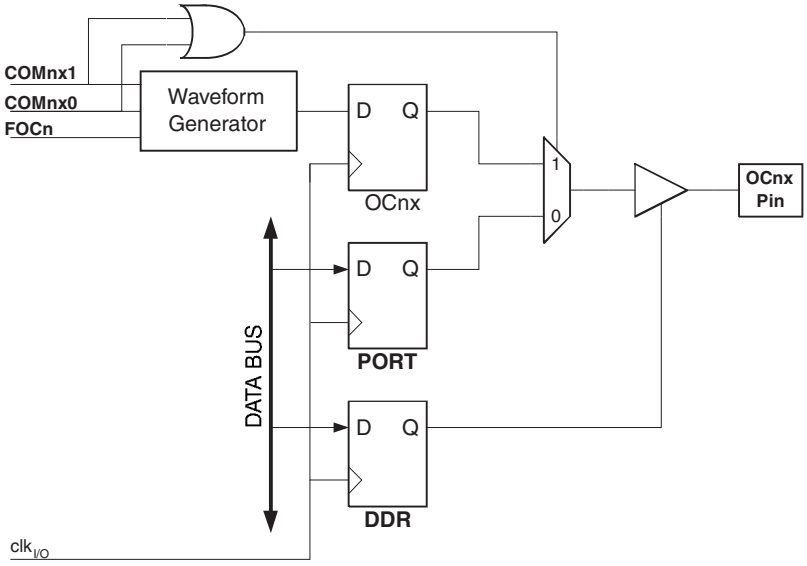
The setup of the OC0x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0x value is to use the Force Output Compare (FOC0x) strobe bits in Normal mode. The OC0x Registers keep their values even when changing between Waveform Generation modes.

Be aware that the COM0x1:0 bits are not double buffered together with the compare value. Changing the COM0x1:0 bits will take effect immediately.

15.6 Compare Match Output Unit

The Compare Output mode (COM0x1:0) bits have two functions. The Waveform Generator uses the COM0x1:0 bits for defining the Output Compare (OC0x) state at the next compare match. Also, the COM0x1:0 bits control the OC0x pin output source. Figure 15-4 shows a simplified schematic of the logic affected by the COM0x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM0x1:0 bits are shown. When referring to the OC0x state, the reference is for the internal OC0x Register, not the OC0x pin. If a system reset occur, the OC0x Register is reset to “0”.

Figure 15-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0x) from the Waveform Generator if either of the COM0x1:0 bits are set. However, the OC0x pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0x pin (DDR_OC0x) must be set as output before the OC0x value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initialization of the OC0x state before the output is enabled. Note that some COM0x1:0 bit settings are reserved for certain modes of operation. See ”Register Description” on page 105.

15.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM0x1:0 bits differently in Normal, CTC, and PWM modes. For all modes, setting the COM0x1:0 = 0 tells the Waveform Generator that no action on the OC0x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 15-2 on page 105. For fast PWM mode, refer to Table 15-3 on page 105, and for phase correct PWM refer to Table 15-4 on page 106.

A change of the COM0x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0x strobe bits.

15.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM02:0) and Compare Output mode (COM0x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM0x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM0x1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See ”Compare Match Output Unit” on page 98).

For detailed timing information refer to ”Timer/Counter Timing Diagrams” on page 103.

15.7.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM02:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 becomes zero. The TOV0 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

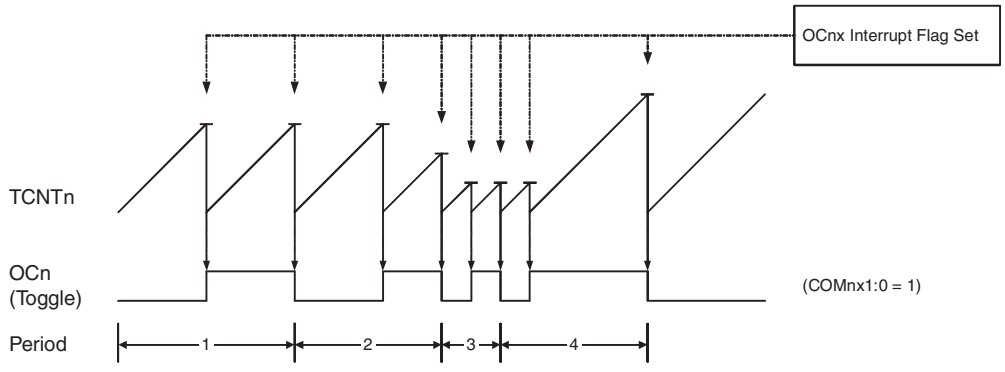
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

15.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM02:0 = 2), the OCR0A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches the OCR0A. The OCR0A defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 15-5. The counter value (TCNT0) increases until a compare match occurs between TCNT0 and OCR0A, and then counter (TCNT0) is cleared.

Figure 15-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC0A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM0A1:0 = 1). The OC0A value will

not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

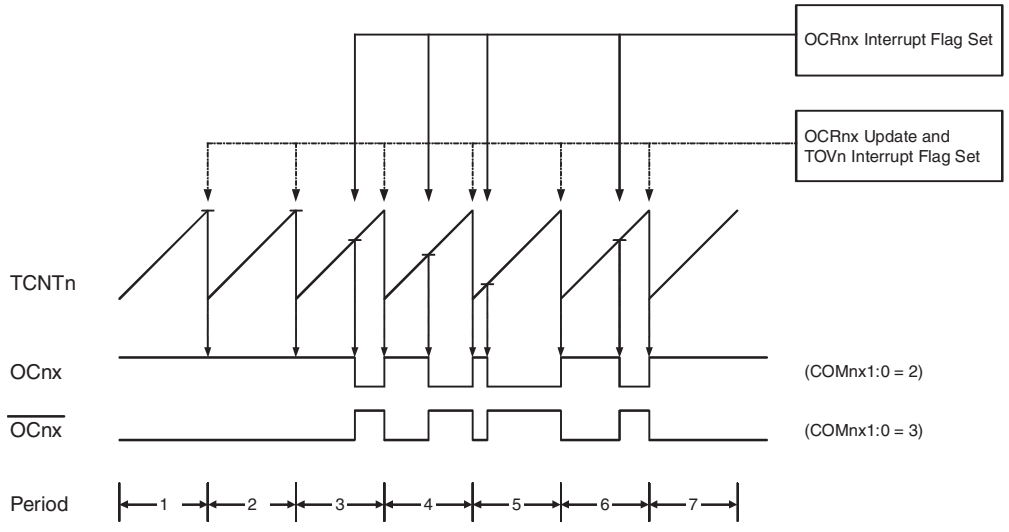
As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

15.7.3 Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM02:0 = 3 or 7) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR0A when WGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 15-6. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

Figure 15-6. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches TOP. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A1:0 bits to one allows the OC0A pin to toggle on Compare Matches if

the WGM02 bit is set. This option is not available for the OC0B pin (see Table 15-6 on page 106). The actual OC0x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0x Register at the compare match between OCR0x and TCNT0, and clearing (or setting) the OC0x Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

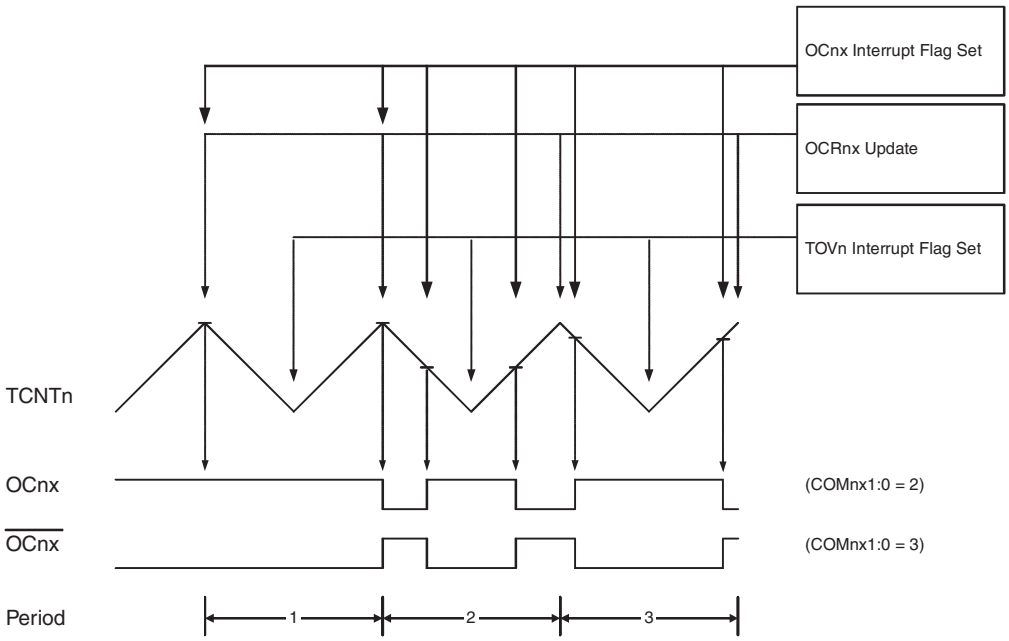
A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0x to toggle its logical level on each compare match (COM0x1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

15.7.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM02:0 = 1 or 5) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1, and OCR0A when WGM2:0 = 5. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 15-7. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0x and TCNT0.

Figure 15-7. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generation of PWM waveforms on the OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A0 bits to one allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see Table 15-7 on page 107). The actual OC0x value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by clearing (or setting) the OC0x Register at the compare match between OCR0x and TCNT0 when the counter increments, and setting (or clearing) the OC0x Register at compare match between OCR0x and TCNT0 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR0A is set equal to BOTTOM, the output will be continuously low and if set equal to MAX the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

At the very start of period 2 in Figure 15-7 OCnx has a transition from high to low even though there is no Compare Match. The point of this transition is to guarantee symmetry around BOTTOM. There are two cases that give a transition without Compare Match.

- OCRnx changes its value from MAX, like in Figure 15-7. When the OCR0A value is MAX the OCn pin value is the same as the result of a down-counting Compare Match. To ensure symmetry around BOTTOM the OCnx value at MAX must correspond to the result of an up-counting Compare Match.
- The timer starts counting from a value higher than the one in OCRnx, and for that reason misses the Compare Match and hence the OCnx change that would have happened on the way up.

15.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when interrupt flags are set. Figure 15-8 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 15-8. Timer/Counter Timing Diagram, no Prescaling

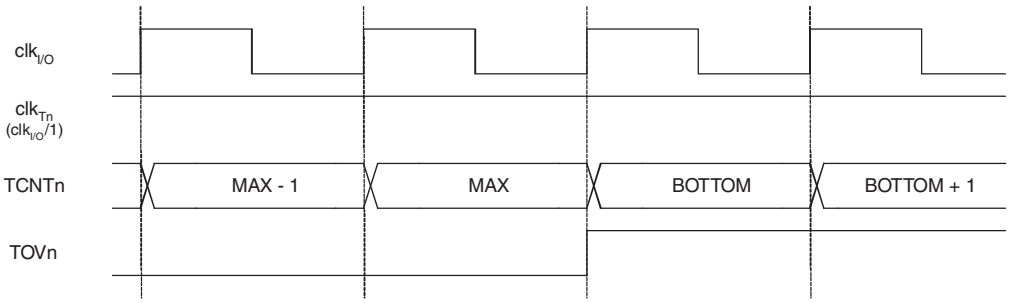


Figure 15-9 shows the same timing data, but with the prescaler enabled.

Figure 15-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

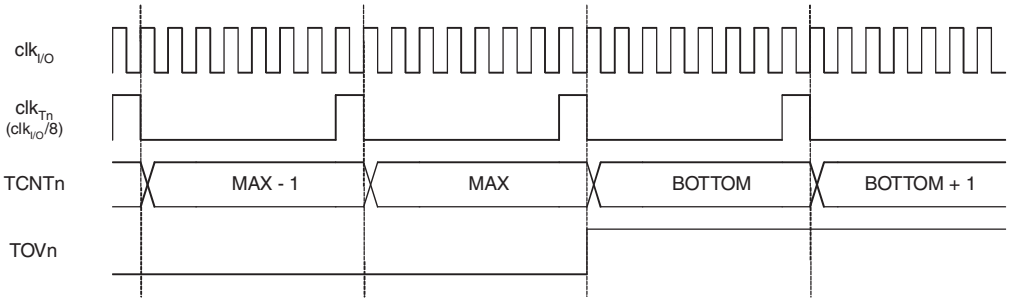


Figure 15-10 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 15-10. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk\_I/O}/8$)

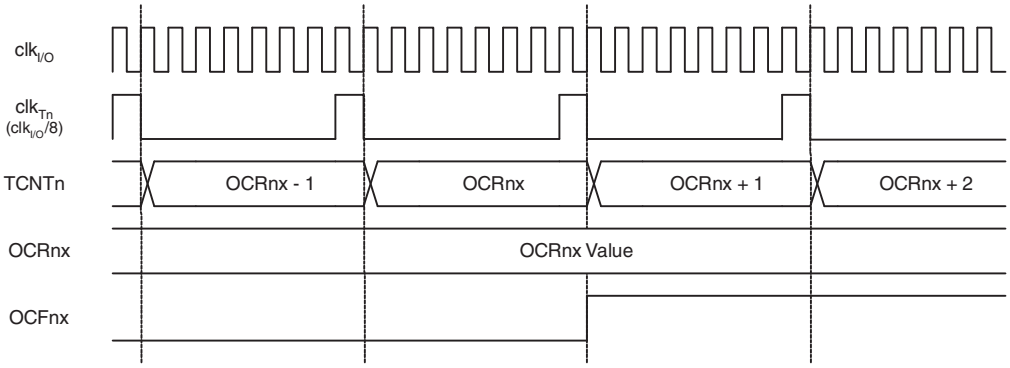
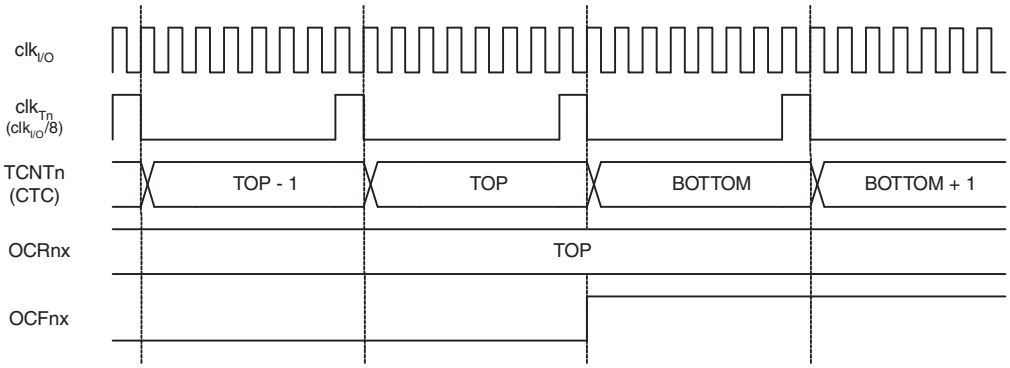


Figure 15-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 15-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}/8$)



15.9 Register Description

15.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 15-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 15-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Table 15-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 15-3. Compare Output Mode, Fast PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See ”Fast PWM Mode” on page 100 for more details.

Table 15-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 15-4. Compare Output Mode, Phase Correct PWM Mode[1]

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See ”Phase Correct PWM Mode” on page 126 for more details.

- **Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 15-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 15-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Table 15-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 15-6. Compare Output Mode, Fast PWM Mode[1]

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at BOTTOM, (non-inverting mode)
1	1	Set OC0B on Compare Match, clear OC0B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See ”Fast PWM Mode” on page 100 for more details.

Table 15-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 15-7. Compare Output Mode, Phase Correct PWM Mode[1]

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match when up-counting. Set OC0B on Compare Match when down-counting.
1	1	Set OC0B on Compare Match when up-counting. Clear OC0B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See ”Phase Correct PWM Mode” on page 101 for more details.

- **Bits 3, 2 – Reserved**

These bits are reserved bits in the ATmega48A/PA/88A/PA/168A/PA/328/P and will always read as zero.

- **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 15-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see ”Modes of Operation” on page 99).

Table 15-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[1][2]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

15.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B: Force Output Compare B**

The FOC0B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- **Bits 5:4 – Reserved**

These bits are reserved bits in the ATmega48A/PA/88A/PA/168A/PA/328/P and will always read as zero.

- **Bit 3 – WGM02: Waveform Generation Mode**

See the description in the ”TCCR0A – Timer/Counter Control Register A” on page 105.

- **Bits 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter.

Table 15-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

15.9.3 TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0x Registers.

15.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0A pin.

15.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0B pin.

15.9.6 TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Reserved**

These bits are reserved bits in the ATmega48A/PA/88A/PA/168A/PA/328/P and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR0.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

15.9.7 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Reserved**

These bits are reserved bits in the ATmega48A/PA/88A/PA/168A/PA/328/P and will always read as zero.

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**

The OCF0B bit is set when a Compare Match occurs between the Timer/Counter and the data in OCR0B – Output Compare Register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable), and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 Compare Match Interrupt Enable), and OCF0A are set, the Timer/Counter0 Compare Match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the

SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed.

The setting of this flag is dependent of the WGM02:0 bit setting. Refer to Table 15-8, ”Waveform Generation Mode Bit Description” on page 107.

19. SPI – Serial Peripheral Interface

19.1 Features

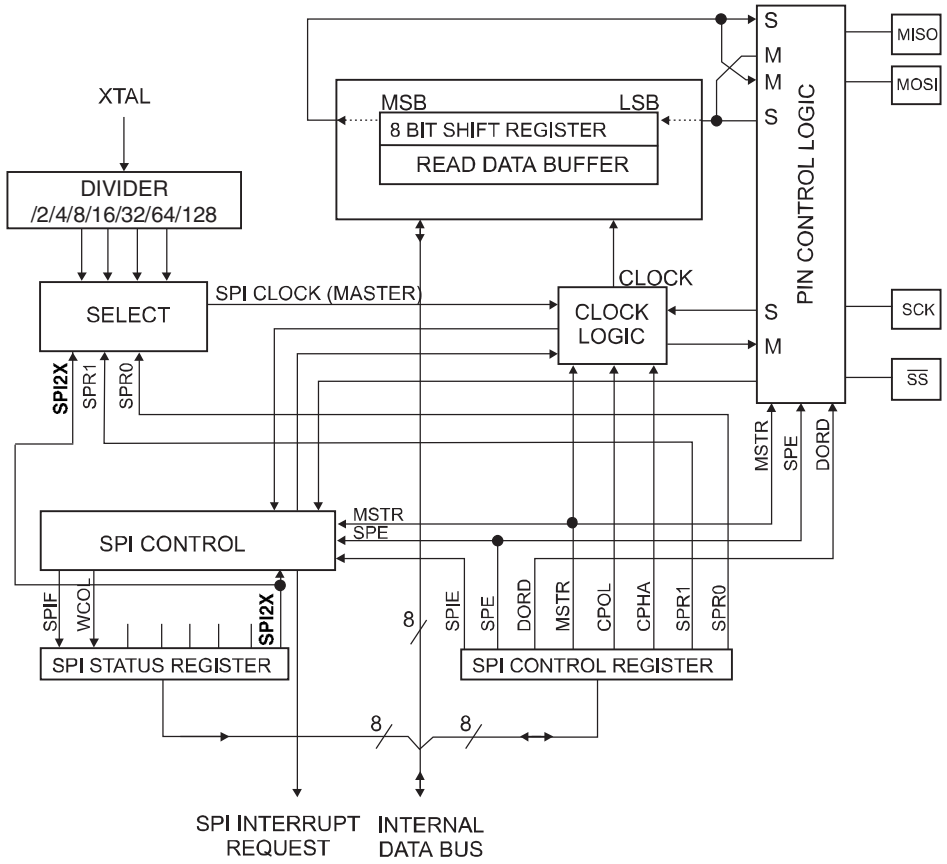
- **Full-duplex, Three-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Seven Programmable Bit Rates**
- **End of Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode**
- **Double Speed (CK/2) Master SPI Mode**

19.2 Overview

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega48A/PA/88A/PA/168A/PA/328/P and peripheral devices or between several AVR devices.

The USART can also be used in Master SPI mode, see “USART in SPI Mode” on page 199. The PRSPI bit in ”Minimizing Power Consumption” on page 41 must be written to zero to enable SPI module.

Figure 19-1. SPI Block Diagram[1]



Note: 1. Refer to Figure 1-1 on page 2, and Table 14-3 on page 83 for SPI pin placement.

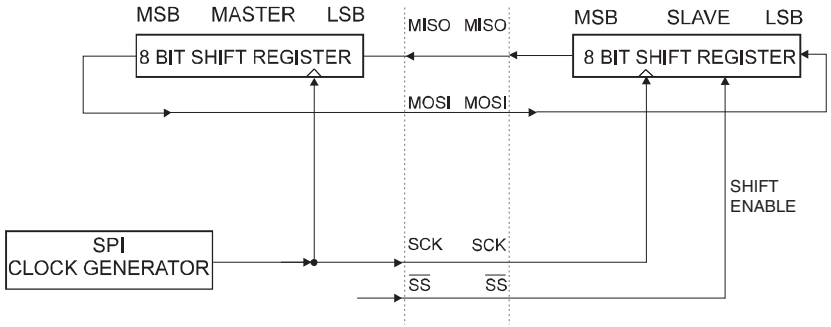
The interconnection between Master and Slave CPUs with SPI is shown in Figure 19-2 on page 163. The system consists of two shift Registers, and a Master clock generator. The SPI Master initiates the communication cycle

when pulling low the Slave Select $\overline{SS}$ pin of the desired Slave. Master and Slave prepare the data to be sent in their respective shift Registers, and the Master generates the required clock pulses on the SCK line to interchange data. Data is always shifted from Master to Slave on the Master Out – Slave In, MOSI, line, and from Slave to Master on the Master In – Slave Out, MISO, line. After each data packet, the Master will synchronize the Slave by pulling high the Slave Select, $\overline{SS}$, line.

When configured as a Master, the SPI interface has no automatic control of the $\overline{SS}$ line. This must be handled by user software before communication can start. When this is done, writing a byte to the SPI Data Register starts the SPI clock generator, and the hardware shifts the eight bits into the Slave. After shifting one byte, the SPI clock generator stops, setting the end of Transmission Flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high the Slave Select, $\overline{SS}$ line. The last incoming byte will be kept in the Buffer Register for later use.

When configured as a Slave, the SPI interface will remain sleeping with MISO tri-stated as long as the $\overline{SS}$ pin is driven high. In this state, software may update the contents of the SPI Data Register, SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the $\overline{SS}$ pin is driven low. As one byte has been completely shifted, the end of Transmission Flag, SPIF is set. If the SPI Interrupt Enable bit, SPIE, in the SPCR Register is set, an interrupt is requested. The Slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte will be kept in the Buffer Register for later use.

Figure 19-2. SPI Master-slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first byte is lost.

In SPI Slave mode, the control logic will sample the incoming signal of the SCK pin. To ensure correct sampling of the clock signal, the minimum low and high periods should be:

Low periods: Longer than 2 CPU clock cycles.

High periods: Longer than 2 CPU clock cycles.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK, and $\overline{SS}$ pins is overridden according to Table 19-1 on page 164. For more details on automatic port overrides, refer to "Alternate Port Functions" on page 81.

Table 19-1. SPI Pin Overrides(Note:)

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

Note: See "Alternate Functions of Port B" on page 83 for a detailed description of how to define the direction of the user defined SPI pins.

The following code examples show how to initialize the SPI as a Master and how to perform a simple transmission. DDR_SPI in the examples must be replaced by the actual Data Direction Register controlling the SPI pins. DD_MOSI, DD_MISO and DD_SCK must be replaced by the actual data direction bits for these pins. E.g. if MOSI is placed on pin PB5, replace DD_MOSI with DDB5 and DDR_SPI with DDRB.

Assembly Code Example[1]

```
SPI_MasterInit:
  ; Set MOSI and SCK output, all others input
  ldi  r17,(1<<DD_MOSI)|(1<<DD_SCK)
  out  DDR_SPI,r17
  ; Enable SPI, Master, set clock rate fck/16
  ldi  r17,(1<<SPE)|(1<<MSTR)|(1<<SPR0)
  out  SPCR,r17
  ret

SPI_MasterTransmit:
  ; Start transmission of data (r16)
  out  SPDR,r16
Wait_Transmit:
  ; Wait for transmission complete
  in   r16, SPSR
  sbrsr16, SPIF
  rjmp Wait_Transmit
  ret
```

C Code Example[1]

```
void SPI_MasterInit(void)
{
  /* Set MOSI and SCK output, all others input */
  DDR_SPI = (1<<DD_MOSI)|(1<<DD_SCK);
  /* Enable SPI, Master, set clock rate fck/16 */
  SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
  /* Start transmission */
  SPDR = cData;
  /* Wait for transmission complete */
  while(!(SPSR & (1<<SPIF)))
    ;
}
```

Note: 1. See "About Code Examples" on page 7.

The following code examples show how to initialize the SPI as a Slave and how to perform a simple reception.

Assembly Code Example[1]

```
SPI_SlaveInit:
  ; Set MISO output, all others input
  ldi   r17,(1<<DD_MISO)
  out   DDR_SPI,r17
  ; Enable SPI
  ldi   r17,(1<<SPE)
  out   SPCR,r17
  ret

SPI_SlaveReceive:
  ; Wait for reception complete
  in r16, SPSR
  sbrs r16, SPIF
  rjmp SPI_SlaveReceive
  ; Read received data and return
  in    r16,SPDR
  ret
```

C Code Example[1]

```
void SPI_SlaveInit(void)
{
  /* Set MISO output, all others input */
  DDR_SPI = (1<<DD_MISO);
  /* Enable SPI */
  SPCR = (1<<SPE);
}

char SPI_SlaveReceive(void)
{
  /* Wait for reception complete */
  while(!(SPSR & (1<<SPIF)))
    ;
  /* Return Data Register */
  return SPDR;
}
```

Note: 1. See ”About Code Examples” on page 7.

19.3 $\overline{SS}$ Pin Functionality

19.3.1 Slave Mode

When the SPI is configured as a Slave, the Slave Select ($\overline{SS}$) pin is always input. When $\overline{SS}$ is held low, the SPI is activated, and MISO becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is driven high.

The $\overline{SS}$ pin is useful for packet/byte synchronization to keep the slave bit counter synchronous with the master clock generator. When the $\overline{SS}$ pin is driven high, the SPI slave will immediately reset the send and receive logic, and drop any partially received data in the Shift Register.

19.3.2 Master Mode

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin.

If $\overline{SS}$ is configured as an output, the pin is a general output pin which does not affect the SPI system. Typically, the pin will be driving the $\overline{SS}$ pin of the SPI Slave.

If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as a Master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled, and the I-bit in SREG is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmission is used in Master mode, and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. If the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI Master mode.

19.4 Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 19-3 and Figure 19-4 on page 168. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing Table 19-3 on page 169 and Table 19-4 on page 169, as done in Table 19-2.

Table 19-2. SPI Modes

SPI Mode	Conditions	Leading Edge	Trailing eDge
0	CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)
1	CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)

Figure 19-3. SPI Transfer Format with CPHA = 0

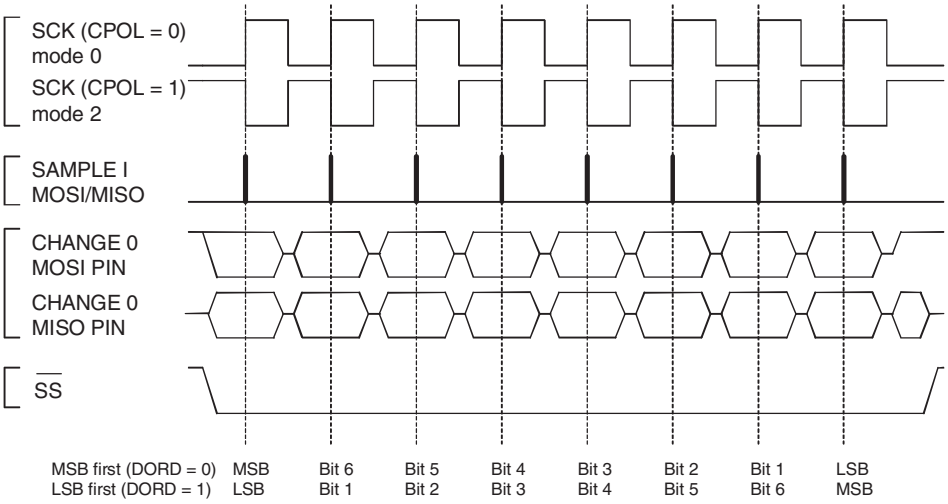
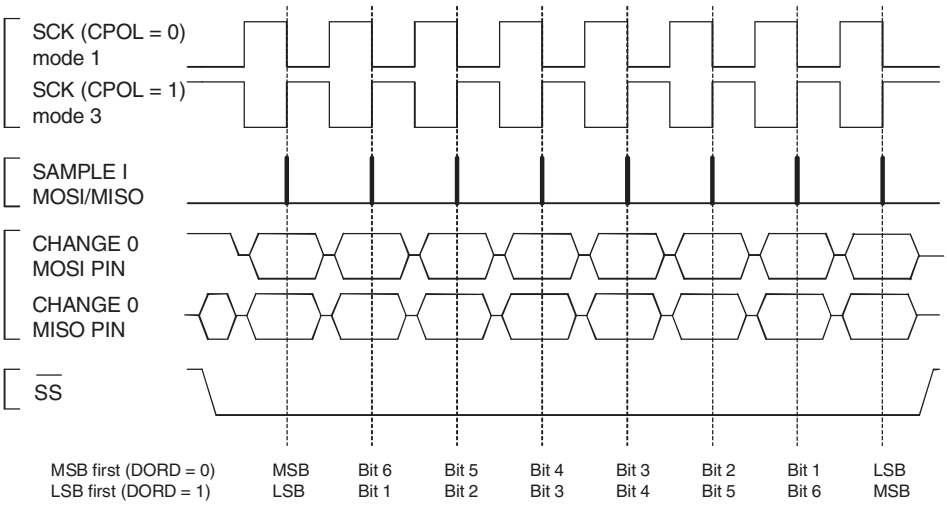


Figure 19-4. SPI Transfer Format with CPHA = 1



19.5 Register Description

19.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR Register is set and the if the Global Interrupt Enable bit in SREG is set.

- **Bit 6 – SPE: SPI Enable**

When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

When the DORD bit is written to one, the LSB of the data word is transmitted first.

When the DORD bit is written to zero, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to Figure 19-3 and Figure 19-4 for an example. The CPOL functionality is summarized below:

Table 19-3. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

- **Bit 2 – CPHA: Clock Phase**

The settings of the Clock Phase bit (CPHA) determine if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to Figure 19-3 and Figure 19-4 for an example. The CPOL functionality is summarized below:

Table 19-4. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 19-5. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

19.5.2 SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF Flag is set. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. If $\overline{SS}$ is an input and is driven low when the SPI is in Master mode, this will also set the SPIF Flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set, then accessing the SPI Data Register (SPDR).

- **Bit 6 – WCOL: Write COLlision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then accessing the SPI Data Register.

- **Bit [5:1] – Reserved**

These bits are reserved bits in the ATmega48A/PA/88A/PA/168A/PA/328/P and will always read as zero.

- **Bit 0 – SPI2X: Double SPI Speed Bit**

When this bit is written logic one the SPI speed (SCK Frequency) will be doubled when the SPI is in Master mode (see Table 19-5). This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{osc}/4$ or lower.

The SPI interface on the ATmega48A/PA/88A/PA/168A/PA/328/P is also used for program memory and EEPROM downloading or uploading. See page 299 for serial programming and verification.

19.5.3 SPDR – SPI Data Register

Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

The SPI Data Register is a read/write register used for data transfer between the Register File and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

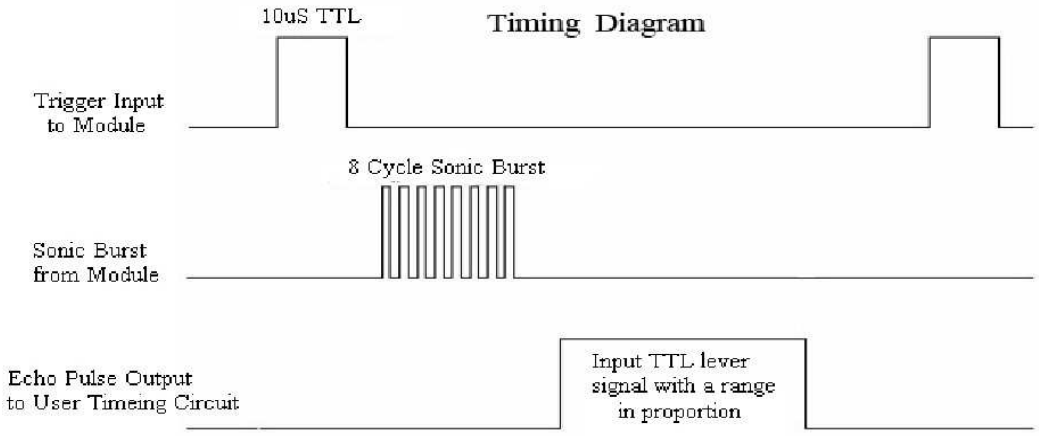
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

ENC28J60
Data Sheet

Stand-Alone Ethernet Controller with SPI™ Interface

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

ENC28J60

Stand-Alone Ethernet Controller with SPI™ Interface

Ethernet Controller Features

- IEEE 802.3 compatible Ethernet controller
- Integrated MAC and 10BASE-T PHY
- Receiver and collision squelch circuit
- Supports one 10BASE-T port with automatic polarity detection and correction
- Supports Full and Half-Duplex modes
- Programmable automatic retransmit on collision
- Programmable padding and CRC generation
- Programmable automatic rejection of erroneous packets
- SPI™ Interface with speeds up to 10 Mb/s

Buffer

- 8-Kbyte transmit/receive packet dual port SRAM
- Configurable transmit/receive buffer size
- Hardware-managed circular receive FIFO
- Byte-wide random and sequential access with auto-increment
- Internal DMA for fast data movement
- Hardware assisted IP checksum calculation

Medium Access Controller (MAC) Features

- Supports Unicast, Multicast and Broadcast packets
- Programmable receive packet filtering and wake-up host on logical AND or OR of the following:
  - Unicast destination address
  - Multicast address
  - Broadcast address
  - Magic Packet™
  - Group destination addresses as defined by 64-bit hash table
  - Programmable pattern matching of up to 64 bytes at user-defined offset
- Loopback mode

Physical Layer (PHY) Features

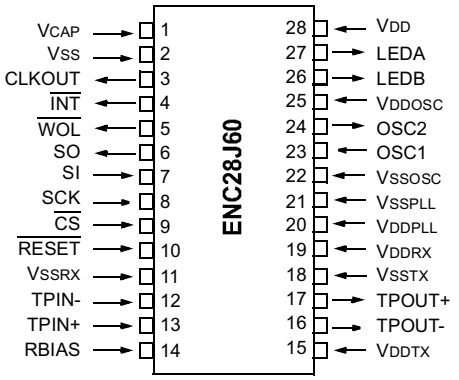
- Wave shaping output filter
- Loopback mode

Operational

- Two programmable LED outputs for LINK, TX, RX, collision and full/half-duplex status
- Seven interrupt sources with two interrupt pins
- 25 MHz clock
- Clock out pin with programmable prescaler
- Operating voltage range of 3.14V to 3.45V
- TTL level inputs
- Temperature range: -40°C to +85°C Industrial, 0°C to +70°C Commercial (SSOP only)
- 28-pin SPDIP, SSOP, SOIC, QFN packages

Package Types

28-Pin SPDIP, SSOP, SOIC



28-pin QFN

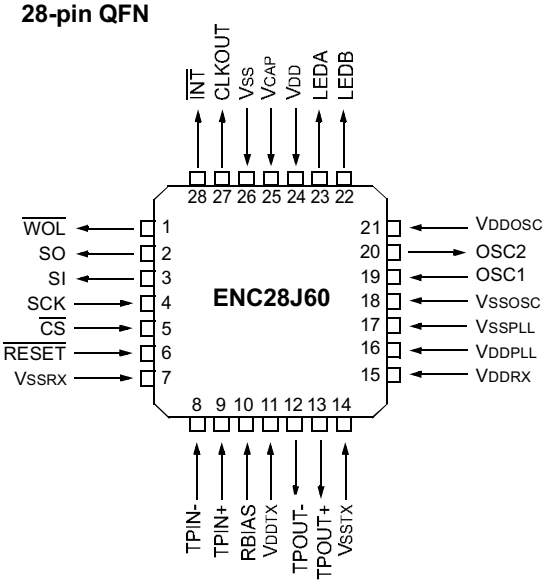


Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 OVERVIEW

The ENC28J60 is a stand-alone Ethernet controller with an industry standard Serial Peripheral Interface (SPI™). It is designed to serve as an Ethernet network interface for any controller equipped with SPI.

The ENC28J60 meets all of the IEEE 802.3 specifications. It incorporates a number of packet filtering schemes to limit incoming packets. It also provides an internal DMA module for fast data throughput and hardware assisted IP checksum calculations. Communication with the host controller is implemented via two interrupt pins and the SPI, with data rates of up to 10 Mb/s. Two dedicated pins are used for LED link and network activity indication.

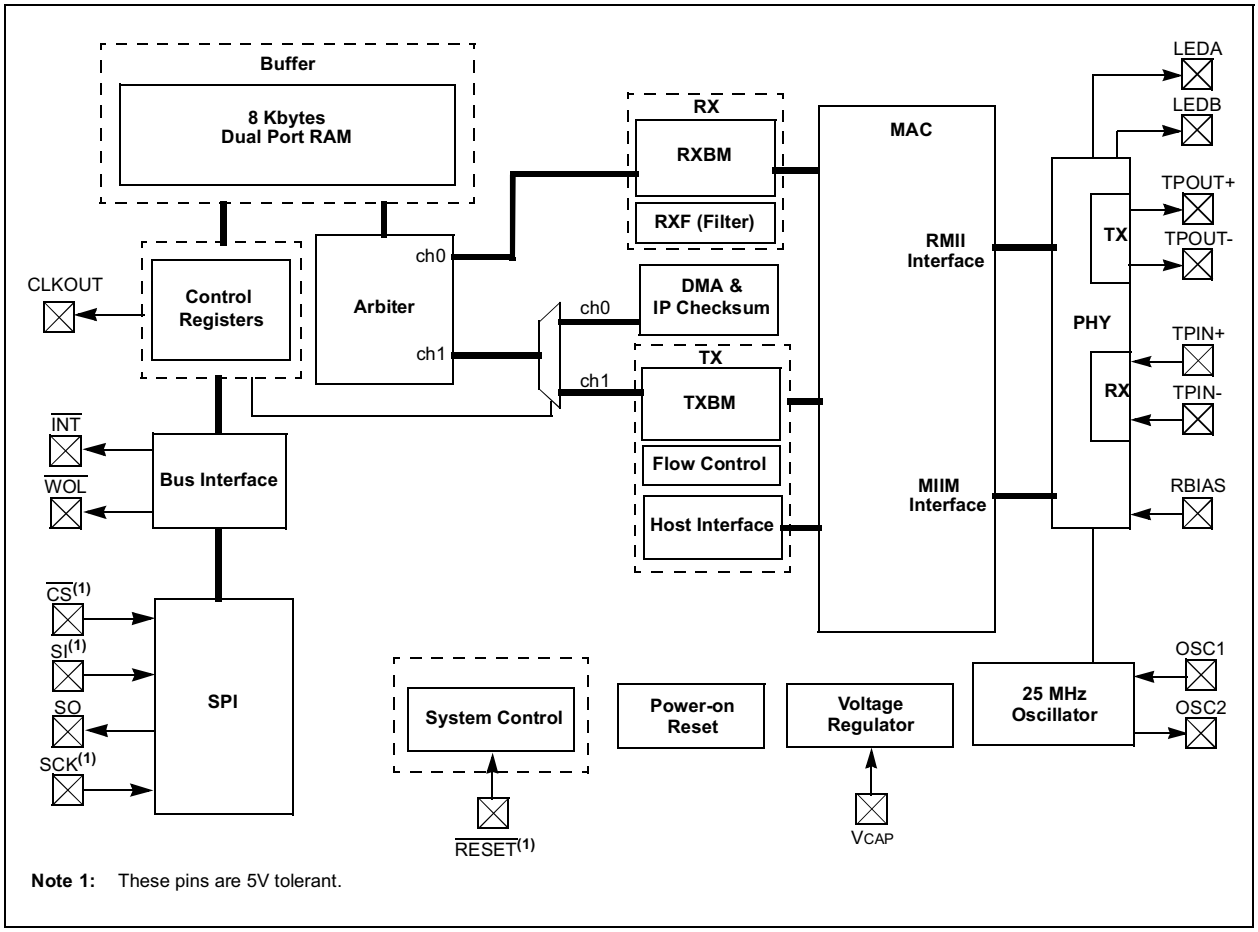
A simple block diagram of the ENC28J60 is shown in Figure 1-1. A typical application circuit using the device is shown in Figure 1-2. With the ENC28J60, two pulse transformers and a few passive components are all that is required to connect a microcontroller to a 10 Mbps Ethernet network.

The ENC28J60 consists of seven major functional blocks:

1. An SPI interface that serves as a communication channel between the host controller and the ENC28J60.
2. Control Registers which are used to control and monitor the ENC28J60.
3. A dual port RAM buffer for received and transmitted data packets.
4. An arbiter to control the access to the RAM buffer when requests are made from DMA, transmit and receive blocks.
5. The bus interface that interprets data and commands received via the SPI interface.
6. The MAC (Medium Access Control) module that implements IEEE 802.3 compliant MAC logic.
7. The PHY (Physical Layer) module that encodes and decodes the analog data that is present on the twisted pair interface.

The device also contains other support blocks, such as the oscillator, on-chip voltage regulator, level translators to provide 5V tolerant I/Os and system control logic.

FIGURE 1-1: ENC28J60 BLOCK DIAGRAM



Note 1: These pins are 5V tolerant.

FIGURE 1-2: TYPICAL ENC28J60-BASED INTERFACE

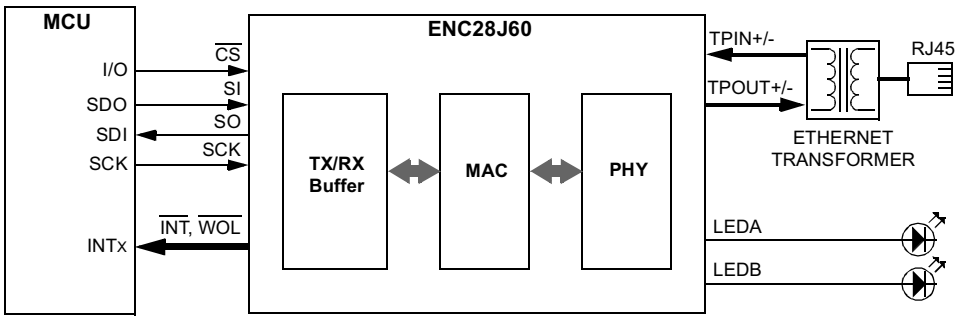


TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	SPDIP, SOIC, SSOP	QFN			
VCAP	1	25	P	—	2.5V output from internal regulator. A 10 µF capacitor to VSSTX must be placed on this pin.
VSS	2	26	P	—	Ground reference.
CLKOUT	3	27	O	—	Programmable clock output pin.(1)
$\overline{INT}$	4	28	O	—	INT interrupt output pin.(2)
$\overline{WOL}$	5	1	O	—	Wake-up on LAN interrupt out pin.(2)
SO	6	2	O	—	Data out pin for SPI™ interface.(2)
SI	7	3	I	ST	Data in pin for SPI interface.(3)
SCK	8	4	I	ST	Clock in pin for SPI interface.(3)
$\overline{CS}$	9	5	I	ST	Chip select input pin for SPI interface.(3,4)
$\overline{RESET}$	10	6	I	ST	Active-low device Reset input.(3, 4)
VSSRX	11	7	P	—	Ground reference for PHY RX.
TPIN-	12	8	I	ANA	Differential signal input.
TPIN+	13	9	I	ANA	Differential signal input.
RBIAS	14	10	I	ANA	Bias current pin for PHY. Must be tied to VSSRX through a 2 kΩ, 1% resistor.
VDDTX	15	11	P	—	Positive supply for PHY TX.
TPOUT-	16	12	O	—	Differential signal output.
TPOUT+	17	13	O	—	Differential signal output.
VSSTX	18	14	P	—	Ground reference for PHY TX.
VDDRX	19	15	P	—	Positive 3.3V supply for PHY RX.
VDDPLL	20	16	P	—	Positive 3.3V supply for PHY PLL.
VSSPLL	21	17	P	—	Ground reference for PHY PLL.
VSSOSC	22	18	P	—	Ground reference for oscillator.
OSC1	23	19	I	DIG	Oscillator input.
OSC2	24	20	O	—	Oscillator output.
VDDOSC	25	21	P	—	Positive 3.3V supply for oscillator.
LEDB	26	22	O	—	LEDB driver pin.(5)
LEDA	27	23	O	—	LEDA driver pin.(5)
VDD	28	24	P	—	Positive 3.3V supply.

Legend: I = Input, O = Output, P = Power, DIG = Digital input, ANA = Analog signal input, ST = Schmitt Trigger

Note 1: Pins have a maximum current capacity of 8 mA.

2: Pins have a maximum current capacity of 4 mA.

3: Pins are 5V tolerant.

4: Pins have an internal weak pull-up to VDD.

5: Pins have a maximum current capacity of 12 mA.

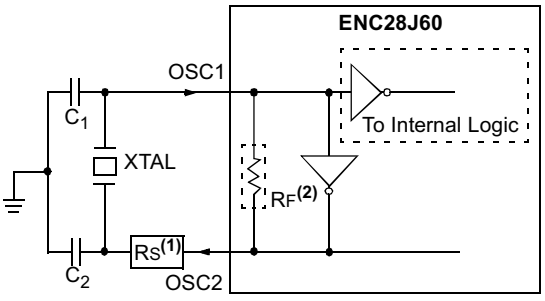
2.0 EXTERNAL CONNECTIONS

2.1 Oscillator

The ENC28J60 is designed to operate at 25 MHz with a crystal connected to the OSC1 and OSC2 pins. The ENC28J60 design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer specifications. A typical oscillator circuit is shown in Figure 2-1.

The ENC28J60 may also be driven by an external clock source connected to the OSC1 pin as shown in Figure 2-2.

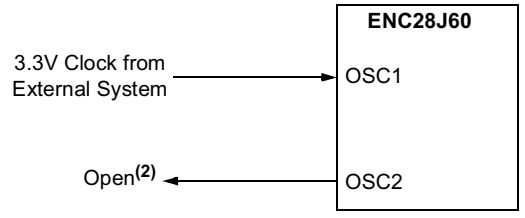
FIGURE 2-1: CRYSTAL OSCILLATOR OPERATION



Note 1: A series resistor, RS, may be required for AT strip cut crystals.

2: The feedback resistor, RF, is typically in the range of 2 to 10 MΩ.

FIGURE 2-2: EXTERNAL CLOCK SOURCE(1)



Note 1: Duty cycle restrictions must be observed.

2: A resistor to ground may be used to reduce system noise. This may increase system current.

2.2 Oscillator Start-up Timer

The ENC28J60 contains an Oscillator Start-up Timer (OST) to ensure that the oscillator and integrated PHY have stabilized before use. The OST does not expire until 7500 OSC1 clock cycles (300 μs) pass after Power-on Reset or wake-up from Power-Down mode occurs. During the delay, all Ethernet registers and buffer memory may still be read and written to through the SPI bus. However, software should not attempt to transmit any packets (set ECON1.TXRTS), enable reception of packets (set ECON1.RXEN) or access any MAC, MII or PHY registers during this period.

When the OST expires, the CLKRDY bit in the ESTAT register will be set. The application software should poll this bit as necessary to determine when normal device operation can begin.

Note: After a Power-on Reset, or the ENC28J60 is removed from Power-Down mode, the CLKRDY bit must be polled before transmitting packets, enabling packet reception or accessing any MAC, MII or PHY registers.

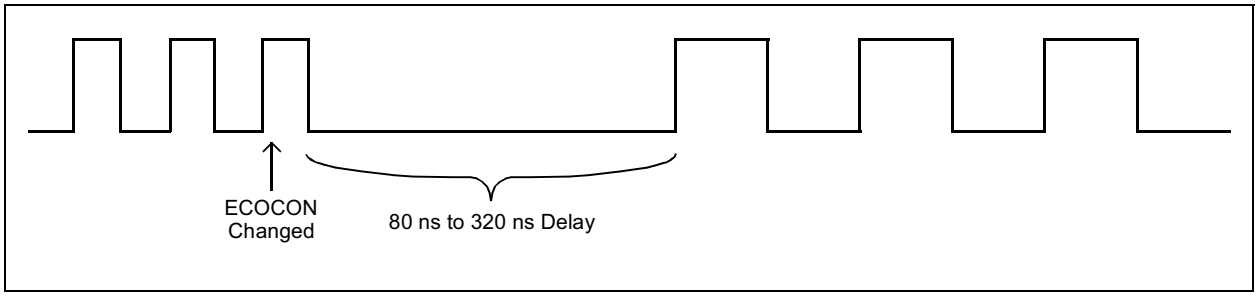
2.3 CLKOUT Pin

The clock out pin is provided to the system designer for use as the host controller clock or as a clock source for other devices in the system. The CLKOUT has an internal prescaler which can divide the output by 1, 2, 3, 4 or 8. The CLKOUT function is enabled and the prescaler is selected via the ECOCON register (Register 2-1).

To create a clean clock signal, the CLKOUT pin is held low for a period when power is first applied. After the Power-on Reset ends, the OST will begin counting. When the OST expires, the CLKOUT pin will begin outputting its default frequency of 6.25 MHz (main clock divided by 4). At any future time that the ENC28J60 is reset by software or the $\overline{\text{RESET}}$ pin, the CLKOUT function will not be altered (ECOCON will not change value). Additionally, Power-Down mode may be entered and the CLKOUT function will continue to operate. When Power-Down mode is cancelled, the OST will be reset but the CLKOUT function will continue. When the CLKOUT function is disabled (ECOCON = 0), the CLKOUT pin is driven low.

The CLKOUT function is designed to ensure that minimum timings are preserved when the CLKOUT pin function is enabled, disabled or the prescaler value is changed. No high or low pulses will be outputted which exceed the frequency specified by the ECOCON configuration. However, when switching frequencies, a delay between two and eight OSC1 clock periods will occur where no clock pulses will be produced (see Figure 2-3). During this period, CLKOUT will be held low.

FIGURE 2-3: CLKOUT TRANSITION



REGISTER 2-1: ECOCON: CLOCK OUTPUT CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	COCON2	COCON1	COCON0
bit 7							bit 0

bit 7-3 **Unimplemented**: Read as '0'

bit 2-0 **COCON2:COCON0**: Clock Output Configuration bits

111 = Reserved for factory test. Do not use. Glitch prevention not assured.
110 = Reserved for factory test. Do not use. Glitch prevention not assured.
101 = CLKOUT outputs main clock divided by 8 (3.125 MHz)
100 = CLKOUT outputs main clock divided by 4 (6.25 MHz)
011 = CLKOUT outputs main clock divided by 3 (8.333333 MHz)
010 = CLKOUT outputs main clock divided by 2 (12.5 MHz)
001 = CLKOUT outputs main clock divided by 1 (25 MHz)
000 = CLKOUT is disabled. The pin is driven low.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.4 Magnetics, Termination and Other External Components

To complete the Ethernet interface, the ENC28J60 requires several standard components to be installed externally. These components should be connected as shown in Figure 2-4.

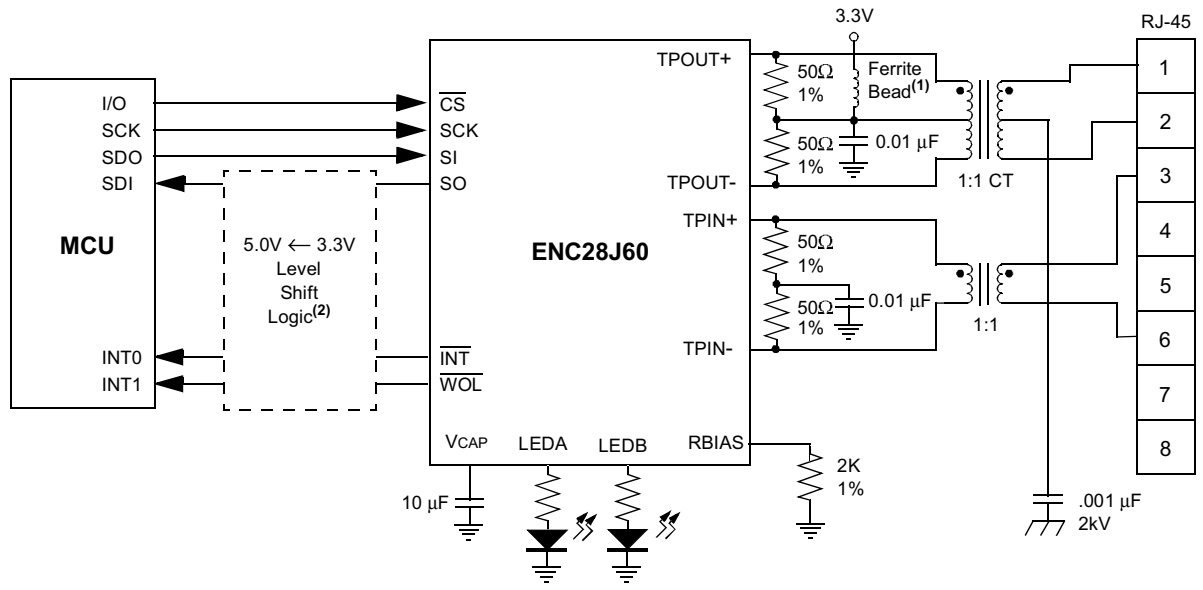
On the differential receive pins (TPIN+/TPIN-), a 1:1 pulse transformer rated for 10BASE-T operation is required. On the differential transmit pins (TPOUT+/TPOUT-), a 1:1 pulse transformer with a center tap is required. The transformers should be rated for isolation of 2 kV or more to protect against static voltages. See **Section 16.0 "Electrical Characteristics"** for specific transformer requirements. Both portions additionally require two 50Ω, 1% resistors and a 0.01 µF capacitor for proper termination.

The internal analog circuitry in the ENC28J60 requires that an external 2 kΩ, 1% resistor be attached from RBIAS to ground.

Some of the digital circuitry in the ENC28J60 operates at a nominal 2.5V to reduce power consumption. A 2.5V regulator is incorporated internally to generate the necessary voltage. The only external component required is a 10 µF capacitor for stability purposes. This capacitor should be attached from VCAP to ground. The internal regulator was not designed to drive external loads.

All power supply pins must be externally connected to the same 3.3V power source. Similarly, all ground references should be externally connected to the same ground node. Each VDD and VSS pin pair should have a 0.1 µF ceramic bypass capacitor placed as close to the pins as possible. Relatively high currents are necessary to operate the twisted pair interface, so all wires should be kept as short as possible and reasonable wire widths should be used on power wires to reduce resistive loss.

FIGURE 2-4: EXTERNAL CONNECTIONS



Note 1: Ferrite Bead should be rated for at least 100 mA.

2: Required only if the microcontroller is operating at 5V.

2.5 I/O Levels

The ENC28J60 is a 3.3V part; however, it was designed to be easily integrated into 5V systems. The SPI $\overline{CS}$, SCK and SI inputs, as well as the $\overline{RESET}$ pin, are all 5V tolerant. On the other hand, if the host controller is operated at 5V, it quite likely will not be within specifications when its SPI and interrupt inputs are driven by the 3.3V CMOS outputs on the ENC28J60. A unidirectional level translator would be necessary.

An economical 74HCT08 (quad AND gate), 74ACT125 (quad 3-state buffer) or many other 5V CMOS chips with TTL level input buffers may be used to provide the necessary level shifting. The use of 3-state buffers permits easy integration into systems which share the SPI bus with other devices. Figure 2-5 and Figure 2-6 show example translation schemes.

FIGURE 2-5: LEVEL SHIFTING USING AND GATES

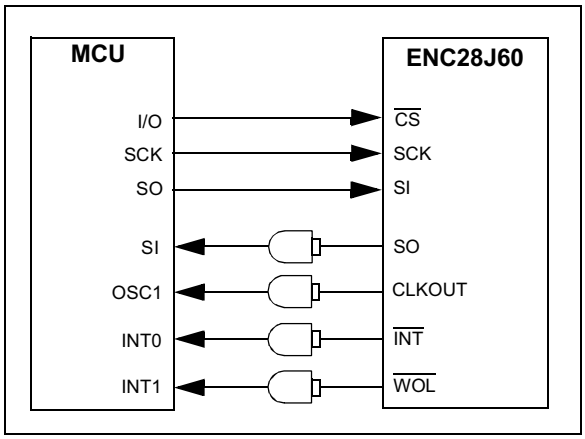
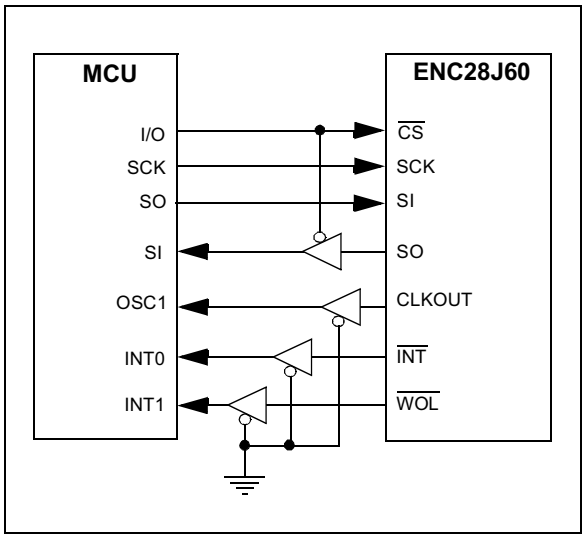


FIGURE 2-6: LEVEL SHIFTING USING 3-STATE BUFFERS

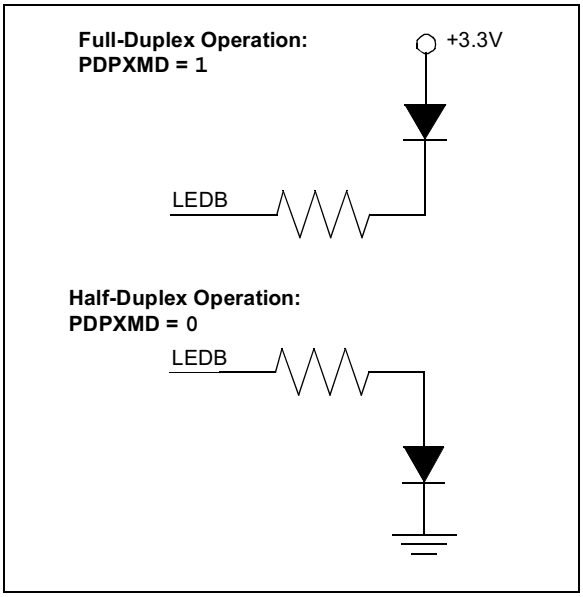


2.6 LED Configuration

The LEDA and LEDB pins support automatic polarity detection on Reset. The LEDs can be connected such that the pin must source current to turn the LED on, or alternately connected such that the pin must sink current to turn the LED on. Upon system Reset, the ENC28J60 will detect how the LED is connected and begin driving the LED to the default state configured by the PHLCON register. If the LED polarity is changed while the ENC28J60 is operating, the new polarity will not be detected until the next system Reset occurs.

LEDB is unique in that the connection of the LED is automatically read on Reset and determines how to initialize the PHCON1.PDPXMD bit. If the pin sources current to illuminate the LED, the bit is cleared on Reset and the PHY defaults to half-duplex operation. If the pin sinks current to illuminate the LED, the bit is set on Reset and the PHY defaults to full-duplex operation. Figure 2-7 shows the two available options. If no LED is attached to the LEDB pin, the PDPXMD bit will reset to an indeterminate value.

FIGURE 2-7: LEDB POLARITY AND RESET CONFIGURATION OPTIONS



REGISTER 2-2: PHLCON: PHY MODULE LED CONTROL REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0
r	r	r	r	LACFG3	LACFG2	LACFG1	LACFG0
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-x
LBCFG3	LBCFG2	LBCFG1	LBCFG0	LFRQ1	LFRQ0	STRCH	r
bit 7							bit 0

bit 15-12 **Reserved:** Write as '0'

bit 11-8 **LACFG3:LACFG0:** LEDA Configuration bits
0000 = Reserved
0001 = Display transmit activity (stretchable)
0010 = Display receive activity (stretchable)
0011 = Display collision activity (stretchable)
0100 = Display link status
0101 = Display duplex status
0110 = Reserved
0111 = Display transmit and receive activity (stretchable)
1000 = On
1001 = Off
1010 = Blink fast
1011 = Blink slow
1100 = Display link status and receive activity (always stretched)
1101 = Display link status and transmit/receive activity (always stretched)
1110 = Display duplex status and collision activity (always stretched)
1111 = Reserved

bit 7-4 **LBCFG3:LBCFG0:** LEDB Configuration bits
0000 = Reserved
0001 = Display transmit activity (stretchable)
0010 = Display receive activity (stretchable)
0011 = Display collision activity (stretchable)
0100 = Display link status
0101 = Display duplex status
0110 = Reserved
0111 = Display transmit and receive activity (stretchable)
1000 = On
1001 = Off
1010 = Blink fast
1011 = Blink slow
1100 = Display link status and receive activity (always stretched)
1101 = Display link status and transmit/receive activity (always stretched)
1110 = Display duplex status and collision activity (always stretched)
1111 = Reserved

bit 3-2 **LFRQ1:LFRQ0:** LED Pulse Stretch Time Configuration bits
11 = Reserved
10 = Stretch LED events to approximately 139 ms
01 = Stretch LED events to approximately 73 ms
00 = Stretch LED events to approximately 40 ms

bit 1 **STRCH:** LED Pulse Stretching Enable bit
1 = Stretchable LED events will cause lengthened LED pulses based on the LFRQ configuration
0 = Stretchable LED events will only be displayed while they are occurring

bit 0 **Reserved:** Write as '0'

Legend:			
R = Readable bit	W = Writable bit	r = Reserved bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

NOTES:

3.0 MEMORY ORGANIZATION

All memory in the ENC28J60 is implemented as static RAM. There are three types of memory in the ENC28J60:

- Control Registers
- Ethernet Buffer
- PHY Registers

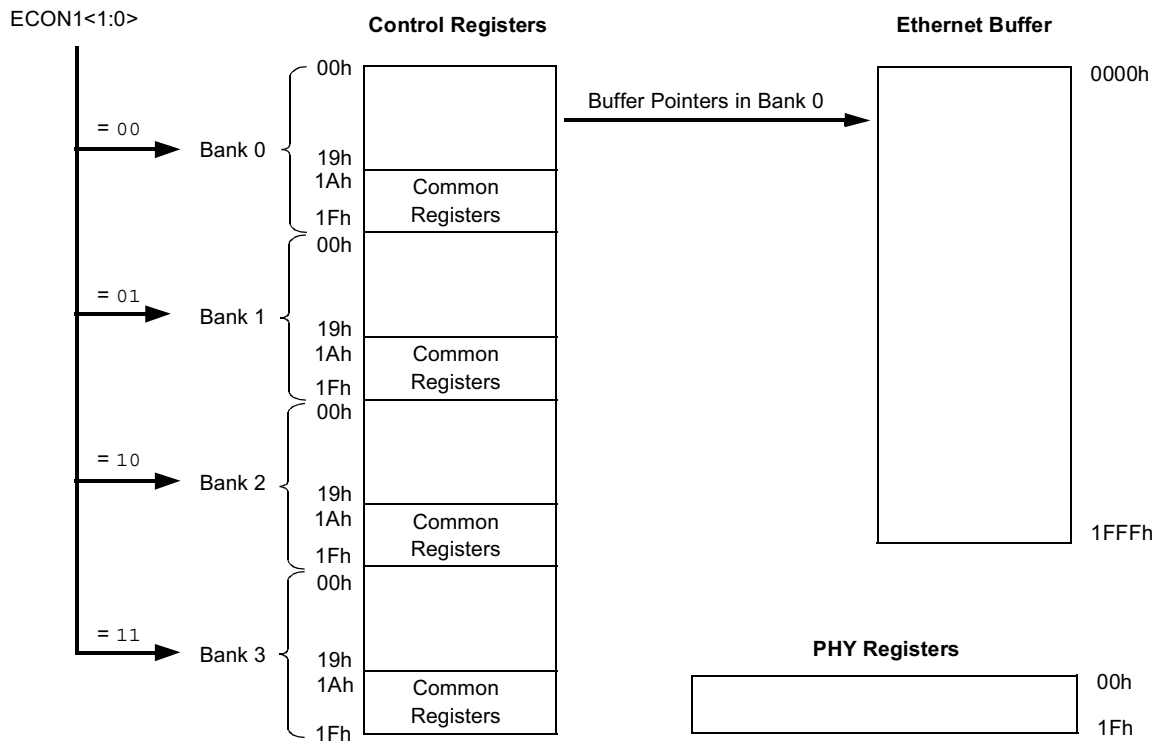
The control registers' memory contains Control Registers (CRs). These are used for configuration, control and status retrieval of the ENC28J60. The Control Registers are directly read and written to by the SPI interface.

The Ethernet buffer contains transmit and receive memory used by the Ethernet controller in a single memory space. The sizes of the memory areas are programmable by the host controller using the SPI interface. The Ethernet buffer memory can only be accessed via the read buffer memory and write buffer memory SPI commands (see **Section 4.2.2 "Read Buffer Memory Command"** and **Section 4.2.4 "Write Buffer Memory Command"**).

The PHY registers are used for configuration, control and status retrieval of the PHY module. The registers are not directly accessible through the SPI interface; they can only be accessed through the Media Independent Interface (MII) implemented in the MAC.

Figure 3-1 shows the data memory organization for the ENC28J60.

FIGURE 3-1: ENC28J60 MEMORY ORGANIZATION



Note: Memory areas are not shown to scale. The size of the control memory space has been scaled to show detail.

3.1 Control Registers

The Control Registers provide the main interface between the host controller and the on-chip Ethernet controller logic. Writing to these registers controls the operation of the interface, while reading the registers allows the host controller to monitor operations.

The Control Register memory is partitioned into four banks, selectable by the bank select bits BSEL1:BSEL0 in the ECON1 register. Each bank is 32 bytes long and addressed by a 5-bit address value.

The last five locations (1Bh to 1Fh) of all banks point to a common set of registers: EIE, EIR, ESTAT, ECON2 and ECON1. These are key registers used in controlling and monitoring the operation of the device. Their common mapping allows easy access without switching the bank. The ECON1 and ECON2 registers are discussed later in this section.

Some of the available addresses are unimplemented. Any attempts to write to these locations are ignored while reads return '0's. The register at address 1Ah in each bank is reserved; read and write operations should not be performed on this register. All other reserved registers may be read, but their contents must not be changed. When reading and writing to registers which contain reserved bits, any rules stated in the register definition should be observed.

Control registers for the ENC28J60 are generically grouped as ETH, MAC and MII registers. Register names starting with "E" belong to the ETH group. Similarly, registers names starting with "MA" belong to the MAC group and registers prefixed with "MI" belong to the MII group.

TABLE 3-1: ENC28J60 CONTROL REGISTER MAP

Bank 0		Bank 1		Bank 2		Bank 3	
Address	Name	Address	Name	Address	Name	Address	Name
00h	ERDPTL	00h	EHT0	00h	MACON1	00h	MAADR1
01h	ERDPTH	01h	EHT1	01h	MACON2	01h	MAADR0
02h	EWRPTL	02h	EHT2	02h	MACON3	02h	MAADR3
03h	EWRPTH	03h	EHT3	03h	MACON4	03h	MAADR2
04h	ETXSTL	04h	EHT4	04h	MABBIPG	04h	MAADR5
05h	ETXSTH	05h	EHT5	05h	—	05h	MAADR4
06h	ETXNDL	06h	EHT6	06h	MAIPGL	06h	EBSTSD
07h	ETXNDH	07h	EHT7	07h	MAIPGH	07h	EBSTCON
08h	ERXSTL	08h	EPMM0	08h	MACLCON1	08h	EBSTCSL
09h	ERXSTH	09h	EPMM1	09h	MACLCON2	09h	EBSTCSH
0Ah	ERXNDL	0Ah	EPMM2	0Ah	MAMXFLL	0Ah	MISTAT
0Bh	ERXNDH	0Bh	EPMM3	0Bh	MAMXFLH	0Bh	—
0Ch	ERXRDPTL	0Ch	EPMM4	0Ch	Reserved	0Ch	—
0Dh	ERXRDPTH	0Dh	EPMM5	0Dh	MAPHSUP	0Dh	—
0Eh	ERXWRPTL	0Eh	EPMM6	0Eh	Reserved	0Eh	—
0Fh	ERXWRPTH	0Fh	EPMM7	0Fh	—	0Fh	—
10h	EDMASTL	10h	EPMCSL	10h	Reserved	10h	—
11h	EDMASTH	11h	EPMCSH	11h	MICON	11h	—
12h	EDMANDL	12h	—	12h	MICMD	12h	EREVID
13h	EDMANDH	13h	—	13h	—	13h	—
14h	EDMADSTL	14h	EPMOL	14h	MIREGADR	14h	—
15h	EDMADSTH	15h	EPMOH	15h	Reserved	15h	ECOCON
16h	EDMACSL	16h	EWOLIE	16h	MIWRL	16h	Reserved
17h	EDMACSH	17h	EWOLIR	17h	MIWRH	17h	EFLOCON
18h	—	18h	ERXFCON	18h	MIRDL	18h	EPAUSL
19h	—	19h	EPKTCNT	19h	MIRDH	19h	EPAUSH
1Ah	Reserved	1Ah	Reserved	1Ah	Reserved	1Ah	Reserved
1Bh	EIE	1Bh	EIE	1Bh	EIE	1Bh	EIE
1Ch	EIR	1Ch	EIR	1Ch	EIR	1Ch	EIR
1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT
1Eh	ECON2	1Eh	ECON2	1Eh	ECON2	1Eh	ECON2
1Fh	ECON1	1Fh	ECON1	1Fh	ECON1	1Fh	ECON1

TABLE 3-2: ENC28J60 CONTROL REGISTER SUMMARY

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Reset	Details on Page
EIE	INTIE	PKTIE	DMAIE	LINKIE	TXIE	WOLIE	TXERIE	RXERIE	0000 0000	67
EIR	—	PKTIF	DMAIF	LINKIF	TXIF	WOLIF	TXERIF	RXERIF	-000 0000	68
ESTAT	INT	r	r	LATECOL	—	RXBUSY	TXABRT	CLKRDY(1)	0000 -000	66
ECON2	AUTOINC	PKTDEC	PWRSV	—	VRPS	—	—	—	100- 0---	16
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	BSEL1	BSEL0	0000 0000	15
ERDPTL	Read Pointer Low Byte ERDPT<7:0>)								1111 1010	17
ERDPTH	—	—	—	Read Pointer High Byte (ERDPT<12:8>)					---0 0101	17
EWRPTL	Write Pointer Low Byte (EWRPT<7:0>)								0000 0000	17
EWRPTH	—	—	—	Write Pointer High Byte (EWRPT<12:8>)					---0 0000	17
ETXSTL	TX Start Low Byte (ETXST<7:0>)								0000 0000	17
ETXSTH	—	—	—	TX Start High Byte (ETXST<12:8>)					---0 0000	17
ETXNDL	TX End Low Byte (ETXND<7:0>)								0000 0000	17
ETXNDH	—	—	—	TX End High Byte (ETXND<12:8>)					---0 0000	17
ERXSTL	RX Start Low Byte (ERXST<7:0>)								1111 1010	17
ERXSTH	—	—	—	RX Start High Byte (ERXST<12:8>)					---0 0101	17
ERXNDL	RX End Low Byte (ERXND<7:0>)								1111 1111	17
ERXNDH	—	—	—	RX End High Byte (ERXND<12:8>)					---1 1111	17
ERXRDPTL	RX RD Pointer Low Byte (ERXRDPT<7:0>)								1111 1010	17
ERXRDPTH	—	—	—	RX RD Pointer High Byte (ERXRDPT<12:8>)					---0 0101	17
ERXWRPTL	RX WR Pointer Low Byte (ERXWRPT<7:0>)								0000 0000	17
ERXWRPTH	—	—	—	RX WR Pointer High Byte (ERXWRPT<12:8>)					---0 0000	17
EDMASTL	DMA Start Low Byte (EDMAST<7:0>)								0000 0000	75
EDMASTH	—	—	—	DMA Start High Byte (EDMAST<12:8>)					---0 0000	75
EDMANDL	DMA End Low Byte (EDMAND<7:0>)								0000 0000	75
EDMANDH	—	—	—	DMA End High Byte (EDMAND<12:8>)					---0 0000	75
EDMADSTL	DMA Destination Low Byte (EDMADST<7:0>)								0000 0000	75
EDMADSTH	—	—	—	DMA Destination High Byte (EDMADST<12:8>)					---0 0000	75
EDMACSL	DMA Checksum Low Byte (EDMACS<7:0>)								0000 0000	76
EDMACSH	DMA Checksum High Byte (EDMACS<15:8>)								0000 0000	76
EHT0	Hash Table Byte 0 (EHT<7:0>)								0000 0000	52
EHT1	Hash Table Byte 1 (EHT<15:8>)								0000 0000	52
EHT2	Hash Table Byte 2 (EHT<23:16>)								0000 0000	52
EHT3	Hash Table Byte 3 (EHT<31:24>)								0000 0000	52
EHT4	Hash Table Byte 4 (EHT<39:32>)								0000 0000	52
EHT5	Hash Table Byte 5 (EHT<47:40>)								0000 0000	52
EHT6	Hash Table Byte 6 (EHT<55:48>)								0000 0000	52
EHT7	Hash Table Byte 7 (EHT<63:56>)								0000 0000	52
EPMM0	Pattern Match Mask Byte 0 (EPMM<7:0>)								0000 0000	51
EPMM1	Pattern Match Mask Byte 1 (EPMM<15:8>)								0000 0000	51
EPMM2	Pattern Match Mask Byte 2 (EPMM<23:16>)								0000 0000	51
EPMM3	Pattern Match Mask Byte 3 (EPMM<31:24>)								0000 0000	51
EPMM4	Pattern Match Mask Byte 4 (EPMM<39:32>)								0000 0000	51
EPMM5	Pattern Match Mask Byte 5 (EPMM<47:40>)								0000 0000	51
EPMM6	Pattern Match Mask Byte 6 (EPMM<55:48>)								0000 0000	51
EPMM7	Pattern Match Mask Byte 7 (EPMM<63:56>)								0000 0000	51
EPMCSL	Pattern Match Checksum Low Byte (EPMCS<7:0>)								0000 0000	51
EPMCSH	Pattern Match Checksum High Byte (EPMCS<15:0>)								0000 0000	51

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved, do not modify.

Note 1: CLKRDY resets to '0' on Power-on Reset but is unaffected on all other Resets.

2: EREVID is a read-only register.

3: ECOCON resets to '---- -100' on Power-on Reset and '---- -uuu' on all other Resets.

TABLE 3-2: ENC28J60 CONTROL REGISTER SUMMARY (CONTINUED)

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Reset	Details on Page
EPMOL	Pattern Match Offset Low Byte (EPMO<7:0>)								0000 0000	51
EPMOH	—	—	—	Pattern Match Offset High Byte (EPMO<12:8>)					---0 0000	51
EWOLIE	UCWOLIE	AWOLIE	—	PMWOLIE	MPWOLIE	HTWOLIE	MCWOLIE	BCWOLIE	00-0 0000	72
EWOLIR	UCWOLIF	AWOLIF	—	PMWOLIF	MPWOLIF	HTWOLIF	MCWOLIF	BCWOLIF	00-0 0000	73
ERXFCON	UCEN	ANDOR	CRCEN	PMEN	MPEN	HTEN	MCEN	BCEN	1010 0001	48
EPKTCNT	Ethernet Packet Count								0000 0000	43
MACON1	—	—	—	LOOPBK	TXPAUS	RXPAUS	PASSALL	MARXEN	---0 0000	34
MACON2	MARST	RNDRST	—	—	MARXRST	RFUNRST	MATXRST	TFUNRST	10-- 0000	61
MACON3	PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDRLEN	HFRMEN	FRMLNEN	FULDPX	0000 0000	35
MACON4	—	DEFER	BPEN	NOBKOFF	—	—	LONGPRE	PUREPRE	-000 --00	36
MABBIPG	—	Back-to-Back Inter-Packet Gap (BBIPG<6:0>)							-000 0000	37
MAIPGL	—	Non-Back-to-Back Inter-Packet Gap Low Byte (MAIPGL<6:0>)							-000 0000	34
MAIPGH	—	Non-Back-to-Back Inter-Packet Gap High Byte (MAIPGH<6:0>)							-000 0000	34
MACLCON1	—	—	—	—	Retransmission Maximum (RETMAX<3:0>)				---- 1111	34
MACLCON2	—	—	Collision Window (COLWIN<5:0>)						--11 0111	34
MAMXFLL	Maximum Frame Length Low Byte (MAMXFL<7:0>)								0000 0000	34
MAMXFLH	Maximum Frame Length High Byte (MAMXFL<15:8>)								0000 0110	34
MAPHSUP	RSTINTFC	—	—	r	RSTRMII	—	—	r	0--1 0--0	62
MICON	RSTMII	—	—	—	—	—	—	—	0--- ----	21
MICMD	—	—	—	—	—	—	MIISCAN	MIIRD	---- --00	21
MIREGADR	—	—	—	MII Register Address (MIREGADR<4:0>)					---0 0000	19
MIWRL	MII Write Data Low Byte (MIWR<7:0>)								0000 0000	19
MIWRH	MII Write Data High Byte (MIWR<15:8>)								0000 0000	19
MIRDL	MII Read Data Low Byte (MIRD<7:0>)								0000 0000	19
MIRDH	MII Read Data High Byte(MIRD<15:8>)								0000 0000	19
MAADR1	MAC Address Byte 1 (MAADR<15:8>)								0000 0000	34
MAADR0	MAC Address Byte 0 (MAADR<7:0>)								0000 0000	34
MAADR3	MAC Address Byte 3 (MAADR<31:24>)								0000 0000	34
MAADR2	MAC Address Byte 2(MAADR<23:16>)								0000 0000	34
MAADR5	MAC Address Byte 5 (MAADR<48:41>)								0000 0000	34
MAADR4	MAC Address Byte 4 (MAADR<40:32>)								0000 0000	34
EBSTSD	Built-in Self-Test Fill Seed (EBSTSD<7:0>)								0000 0000	80
EBSTCON	PSV2	PSV1	PSV0	PSEL	TMSEL1	TMSEL0	TME	BISTST	0000 0000	79
EBSTCSL	Built-in Self-Test Checksum Low Byte (EBSTCS<7:0>)								0000 0000	80
EBSTCSH	Built-in Self-Test Checksum High Byte (EBSTCS<15:8>)								0000 0000	80
MISTAT	—	—	—	—	r	NVALID	SCAN	BUSY	---- 0000	22
EREVID(2)	—	—	—	Ethernet Revision ID (EREVID<4:0>)					---q qqqq	22
ECOCON(3)	—	—	—	—	—	COCON2	COCON1	COCON0	---- -100	6
EFLOCON	—	—	—	—	—	FULDPXS	FCEN1	FCEN0	---- -000	56
EPAUSL	Pause Timer Value Low Byte (EPAUS<7:0>)								0000 0000	57
EPAUSH	Pause Timer Value High Byte (EPAUS<15:8>)								0001 0000	57

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved, do not modify.

Note 1: CLKRDY resets to '0' on Power-on Reset but is unaffected on all other Resets.

2: EREVID is a read-only register.

3: ECOCON resets to '---- -100' on Power-on Reset and '---- -uuu' on all other Resets.

3.1.1 ECON1 REGISTER

The ECON1 register, shown in Register 3-1, is used to control the main functions of the ENC28J60. Receive enable, transmit request, DMA control and bank select bits can all be found in ECON1.

REGISTER 3-1: ECON1: ETHERNET CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	BSEL1	BSEL0
bit 7							bit 0

bit 7 **TXRST:** Transmit Logic Reset bit
1 = Transmit logic is held in Reset
0 = Normal operation

bit 6 **RXRST:** Receive Logic Reset bit
1 = Receive logic is held in Reset
0 = Normal operation

bit 5 **DMAST:** DMA Start and Busy Status bit
1 = DMA copy or checksum operation is in progress
0 = DMA hardware is Idle

bit 4 **CSUMEN:** DMA Checksum Enable bit
1 = DMA hardware calculates checksums
0 = DMA hardware copies buffer memory

bit 3 **TXRTS:** Transmit Request To Send bit
1 = The transmit logic is attempting to transmit a packet
0 = The transmit logic is Idle

bit 2 **RXEN:** Receive Enable bit
1 = Packets which pass the current filter configuration will be written into the receive buffer
0 = All packets received will be ignored

bit 1-0 **BSEL1:BSEL0:** Bank Select bits
11 = SPI accesses registers in Bank 3
10 = SPI accesses registers in Bank 2
01 = SPI accesses registers in Bank 1
00 = SPI accesses registers in Bank 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.1.2 ECON2 REGISTER

The ECON2 register, shown in Register 3-2, is used to control other main functions of the ENC28J60.

REGISTER 3-2: ECON2: ETHERNET CONTROL REGISTER 2

R/W-1	W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
AUTOINC	PKTDEC	PWRSV	—	VRPS	—	—	—
bit 7							bit 0

bit 7 **AUTOINC:** Automatic Buffer Pointer Increment Enable bit
1 = Automatically increment ERDPT and EWRPT when the SPI RBM/WBM command is used
0 = Do not automatically change ERDPT and EWRPT after the buffer is accessed

bit 6 **PKTDEC:** Packet Decrement bit
1 = Decrement the EPKTCNT register by one
0 = Leave EPKTCNT unchanged

bit 5 **PWRSV:** Power Save Enable bit
1 = MAC, PHY and control logic are in Low-Power Sleep mode
0 = Normal operation

bit 4 **Unimplemented:** Read as '0'

bit 3 **VRPS:** Voltage Regulator Power Save Enable bit
When PWRSV = 1:
1 = Internal voltage regulator is in Low-Current mode
0 = Internal voltage regulator is in Normal Current mode
When PWRSV = 0:
The bit is ignored; the regulator always outputs as much current as the device requires.

bit 2-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.2 Ethernet Buffer

The Ethernet buffer contains transmit and receive memory used by the Ethernet controller. The entire buffer is 8 Kbytes, divided into separate receive and transmit buffer spaces. The sizes and locations of transmit and receive memory are fully programmable by the host controller using the SPI interface.

The relationship of the buffer spaces is shown in Figure 3-2.

3.2.1 RECEIVE BUFFER

The receive buffer constitutes a circular FIFO buffer managed by hardware. The register pairs ERXSTH:ERXSTL and ERXNDH:ERXNDL serve as pointers to define the buffer's size and location within the memory. The byte pointed to by ERXST and the byte pointed to by ERXND are both included in the FIFO buffer.

As bytes of data are received from the Ethernet interface, they are written into the receive buffer sequentially. However, after the memory pointed to by ERXND is written to, the hardware will automatically write the next byte of received data to the memory pointed to by ERXST. As a result, the receive hardware will never write outside the boundaries of the FIFO.

The host controller may program the ERXST and ERXND pointers when the receive logic is not enabled. The pointers must not be modified while the receive logic is enabled (ECON1.RXEN is set). If desired, the pointers may span the 1FFFh to 0000h memory boundary; the hardware will still operate as a FIFO.

The ERXWRPTH:ERXWRPTL registers define a location within the FIFO where the hardware will write bytes that it receives. The pointer is read-only and is automatically updated by the hardware whenever a new packet is successfully received. The pointer is useful for determining how much free space is available within the FIFO.

The ERXRDPT registers define a location within the FIFO where the receive hardware is forbidden to write to. In normal operation, the receive hardware will write data up to, but not including, the memory pointed to by ERXRDPT. If the FIFO fills up with data and new data continues to arrive, the hardware will not overwrite the previously received data. Instead, the new data will be thrown away and the old data will be preserved. In order to continuously receive new data, the host controller must periodically advance this pointer whenever it finishes processing some, or all, of the old received data.

3.2.2 TRANSMIT BUFFER

Any space within the 8-Kbyte memory, which is not programmed as part of the receive FIFO buffer, is considered to be the transmit buffer. The responsibility of managing where packets are located in the transmit buffer belongs to the host controller. Whenever the host controller decides to transmit a packet, the ETXST and ETXND pointers are programmed with addresses specifying where, within the transmit buffer, the particular packet to transmit is located. The hardware does not check that the start and end addresses do not overlap with the receive buffer. To prevent buffer corruption, the host controller must make sure to not transmit a packet while the ETXST and ETXND pointers are overlapping the receive buffer, or while the ETXND pointer is too close to the receive buffer. See **Section 7.1 "Transmitting Packets"** for more information.

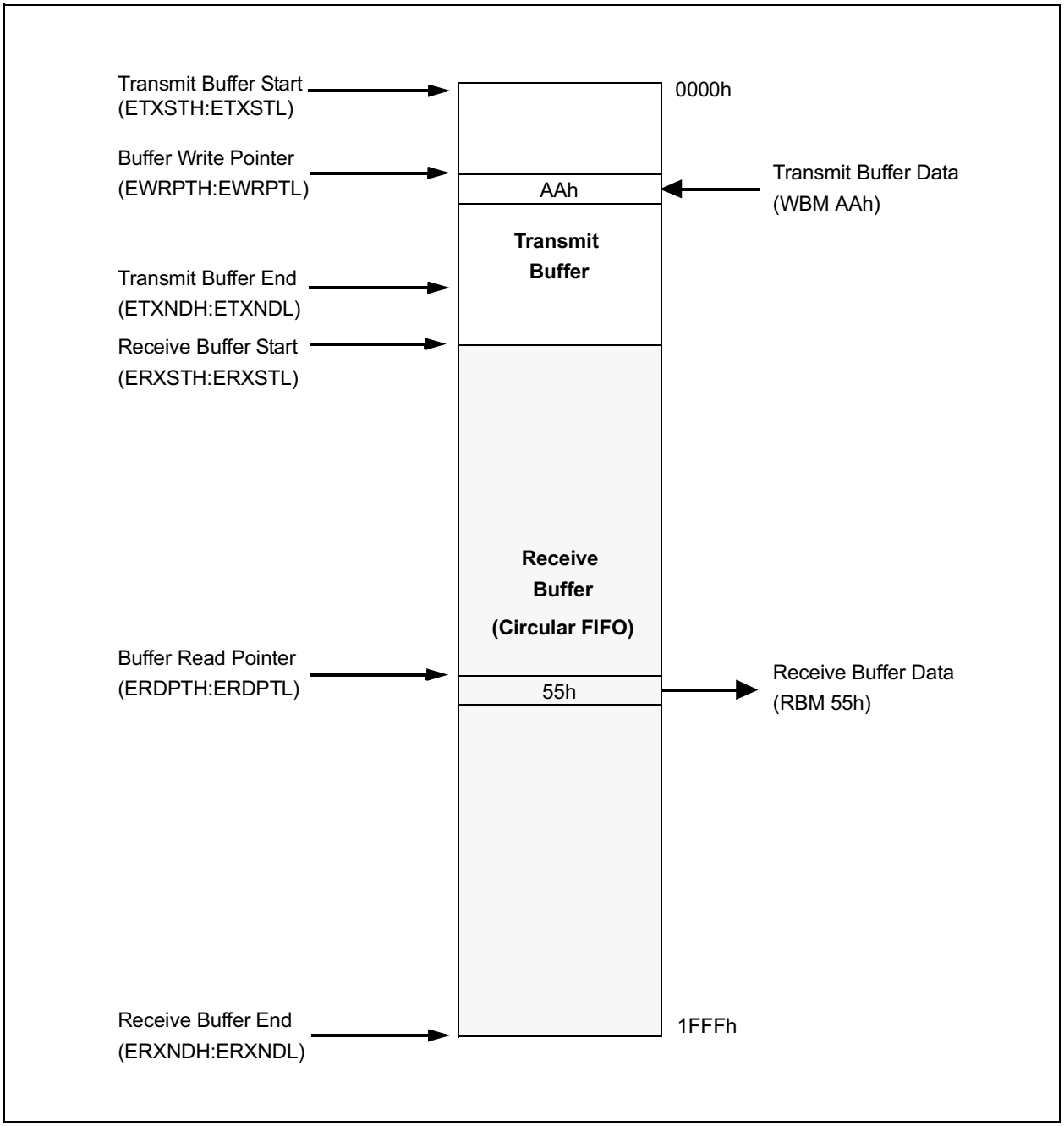
3.2.3 READING AND WRITING TO THE BUFFER

The Ethernet buffer contents are accessed from the host controller though separate read and write pointers (ERDPT and EWRPT) combined with the read buffer memory and write buffer memory SPI commands. While sequentially reading from the receive buffer, a wrapping condition will occur at the end of the receive buffer. While sequentially writing to the buffer, no wrapping conditions will occur. See **Section 4.2.2 "Read Buffer Memory Command"** and **Section 4.2.4 "Write Buffer Memory Command"** for more information.

3.2.4 DMA ACCESS TO THE BUFFER

The integrated DMA controller must read from the buffer when calculating a checksum and it must read and write to the buffer when copying memory. The DMA follows the same wrapping rules that SPI accesses do. While it sequentially reads, it will be subject to a wrapping condition at the end of the receive buffer. All writes it does will not be subject to any wrapping conditions. See **Section 13.0 "Direct Memory Access Controller"** for more information.

FIGURE 3-2: ETHERNET BUFFER ORGANIZATION



3.3 PHY Registers

The PHY registers provide configuration and control of the PHY module, as well as status information about its operation. All PHY registers are 16 bits in width. There are a total of 32 PHY addresses; however, only 9 locations are implemented. Writes to unimplemented locations are ignored and any attempts to read these locations will return '0'. All reserved locations should be written as '0'; their contents should be ignored when read.

Unlike the ETH, MAC and MII control registers, or the buffer memory, the PHY registers are not directly accessible through the SPI control interface. Instead, access is accomplished through a special set of MAC control registers that implement a Media Independent Interface for Management (MIIM). These control registers are referred to as the MII registers. The registers that control access to the PHY registers are shown in Register 3-3 and Register 3-4.

3.3.1 READING PHY REGISTERS

When a PHY register is read, the entire 16 bits are obtained.

To read from a PHY register:

1. Write the address of the PHY register to read from into the MIREGADR register.
2. Set the MICMD.MIIRD bit. The read operation begins and the MISTAT.BUSY bit is set.
3. Wait 10.24 μs. Poll the MISTAT.BUSY bit to be certain that the operation is complete. While busy, the host controller should not start any MIISCAN operations or write to the MIWRH register.
   When the MAC has obtained the register contents, the BUSY bit will clear itself.
4. Clear the MICMD.MIIRD bit.
5. Read the desired data from the MIRDL and MIRDH registers. The order that these bytes are accessed is unimportant.

3.3.2 WRITING PHY REGISTERS

When a PHY register is written to, the entire 16 bits is written at once; selective bit writes are not implemented. If it is necessary to reprogram only select bits in the register, the controller must first read the PHY register, modify the resulting data and then write the data back to the PHY register.

To write to a PHY register:

1. Write the address of the PHY register to write to into the MIREGADR register.
2. Write the lower 8 bits of data to write into the MIWRL register.
3. Write the upper 8 bits of data to write into the MIWRH register. Writing to this register automatically begins the MII transaction, so it must be written to after MIWRL. The MISTAT.BUSY bit becomes set.

The PHY register will be written after the MII operation completes, which takes 10.24 μs. When the write operation has completed, the BUSY bit will clear itself. The host controller should not start any MIISCAN or MIIRD operations while busy.

3.3.3 SCANNING A PHY REGISTER

The MAC can be configured to perform automatic back-to-back read operations on a PHY register. This can significantly reduce the host controller complexity when periodic status information updates are desired. To perform the scan operation:

1. Write the address of the PHY register to read from into the MIREGADR register.
2. Set the MICMD.MIISCAN bit. The scan operation begins and the MISTAT.BUSY bit is set. The first read operation will complete after 10.24 μs. Subsequent reads will be done at the same interval until the operation is cancelled. The MISTAT.NVALID bit may be polled to determine when the first read operation is complete.

After setting the MIISCAN bit, the MIRDL and MIRDH registers will automatically be updated every 10.24 μs. There is no status information which can be used to determine when the MIRD registers are updated. Since the host controller can only read one MII register at a time through the SPI, it must not be assumed that the values of MIRDL and MIRDH were read from the PHY at exactly the same time.

When the MIISCAN operation is in progress, the host controller must not attempt to write to MIWRH or start an MIIRD operation. The MIISCAN operation can be cancelled by clearing the MICMD.MIISCAN bit and then polling the MISTAT.BUSY bit. New operations may be started after the BUSY bit is cleared.

TABLE 3-3: ENC28J60 PHY REGISTER SUMMARY

Addr	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values
00h	PHCON1	PRST	PLOOPBK	—	—	PPWRSV	r	—	PDPXMD(1)	r	—	—	—	—	—	—	—	00-- 10-q 0--- ----
01h	PHSTAT1	—	—	—	PFDPX	PHDPX	—	—	—	—	—	—	—	—	LLSTAT	JBSTAT	—	---1 1--- ---- -00-
02h	PHID1	PHY Identifier (PID18:PID3) = 0083h																0000 0000 1000 0011
03h	PHID2	PHY Identifier (PID24:PID19) = 000101						PHY P/N (PPN5:PPN0) = 00h						PHY Revision (PREV3:PREV0) = 00h				0001 0100 0000 0000
10h	PHCON2	—	FRCLNK	TXDIS	r	r	JABBER	r	HDLDIS	r	r	r	r	r	r	r	r	-000 0000 0000 0000
11h	PHSTAT2	—	—	TXSTAT	RXSTAT	COLSTAT	LSTAT	DPXSTAT(1)	—	—	—	—	PLRITY	—	—	—	—	--00 00q- ---0 ----
12h	PHIE	r	r	r	r	r	r	r	r	r	r	r	PLNKIE	r	r	PGEIE	r	0000 0000 0000 0000
13h	PHIR	r	r	r	r	r	r	r	r	r	r	r	PLNKIF	r	PGIF	r	r	xxxx xxxx xx00 00x0
14h	PHLCON	r	r	r	r	LACFG3:LACFG0				LBCFG3:LBCFG0				LFRQ1:LFRQ0		STRCH	r	0011 0100 0010 001x

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved, do not modify.

Note 1: Reset values of the Duplex mode/status bits depend on the connection of the LED to the LEDB pin (see **Section 2.6 "LED Configuration"** for additional details).

REGISTER 3-3: MICON: MII CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
RSTMII	—	—	—	—	—	—	—
bit 7							bit 0

bit 7 **RSTMII:** MII Management Module Reset bit
1 = MII management module held in Reset
0 = Normal operation

bit 6-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-4: MICMD: MII COMMAND REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MIISCAN	MIIRD
bit 7							bit 0

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **MIISCAN:** MII Scan Enable bit
1 = PHY register at MIREGADR is continously read and the data is placed in MIRD
0 = No MII management scan operation is in progress

bit 0 **MIIRD:** MII Read Enable bit
1 = PHY register at MIREGADR is read once and the data is placed in MIRD
0 = No MII management read operation is in progress

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-5: MISTAT: MII STATUS REGISTER

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	r	NVALID	SCAN	BUSY
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **Reserved:** Maintain '0'

bit 1 **NVALID:** MII Management Read Data Not Valid bit
1 = The contents of MIRD are not valid yet
0 = The MII management read cycle has completed and MIRD has been updated

bit 1 **SCAN:** MII Management Scan Operation bit
1 = MII management scan operation is in progress
0 = No MII management scan operation is in progress

bit 0 **BUSY:** MII Management Busy bit
1 = A PHY register is currently being read or written to
0 = The MII management interface is Idle

Legend:			
R = Readable bit	r = reserved, maintain as '0'	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.3.4 PHSTAT REGISTERS

The PHSTAT1 and PHSTAT2 registers contain read-only bits that show the current status of the PHY module's operations, particularly the conditions of the communications link to the rest of the network.

The PHSTAT1 register (Register 3-6) contains the LLSTAT bit; it clears and latches low if the physical layer link has gone down since the last read of the register. Periodic polling by the host controller can be used to determine exactly when the link fails. It may be particularly useful if the link change interrupt is not used.

The PHSTAT1 register also contains a jabber status bit. An Ethernet controller is said to be "jabbering" if it continuously transmits data without stopping and allowing other nodes to share the medium. Generally, the jabber condition indicates that the local controller may be grossly violating the maximum packet size defined by the IEEE specification. This bit latches high to indicate that a jabber condition has occurred since the last read of the register.

The PHSTAT2 register (Register 3-7) contains status bits which report if the PHY module is linked to the network and whether or not it is transmitting or receiving.

3.3.5 PHID1 AND PHID2 REGISTERS

The PHID1 and PHID2 registers are read-only registers. They hold constant data that help identify the Ethernet controller and may be useful for debugging purposes. This includes:

- The part number of the PHY module (PPN5:PPN0)
- The revision level of the PHY module (PREV3:PREV0); and
- The PHY Identifier, as part of Microchip's corporate Organizationally Unique Identifier (OUI) (PID24:PID3)

The PHY part number and revision are part of PHID2. The upper two bytes of the PHY identifier are located in PHID1, with the remainder in PHID2. The exact locations within registers are shown in Table 3-3.

Revision information is also stored in EREVID. This is a read-only control register which contains a 5-bit identifier for the specific silicon revision level of the device. Details of this register are shown in Table 3-2.

REGISTER 3-6: PHSTAT1: PHYSICAL LAYER STATUS REGISTER 1

U-0	U-0	U-0	R-1	R-1	U-0	U-0	U-0
—	—	—	PFDPX	PHDPX	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/LL-0	R/LH-0	U-0
—	—	—	—	—	LLSTAT	JBRSTAT	—
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **PFDPX:** PHY Full-Duplex Capable bit
1 = PHY is capable of operating at 10 Mbps in Full-Duplex mode (this bit is always set)

bit 11 **PHDPX:** PHY Half-Duplex Capable bit
1 = PHY is capable of operating at 10 Mbps in Half-Duplex mode (this bit is always set)

bit 10-3 **Unimplemented:** Read as '0'

bit 2 **LLSTAT:** PHY Latching Link Status bit
1 = Link is up and has been up continously since PHSTAT1 was last read
0 = Link is down or was down for a period since PHSTAT1 was last read

bit 1 **JBRSTAT:** PHY Latching Jabber Status bit
1 = PHY has detected a transmission meeting the jabber criteria since PHYSTAT1 was last read
0 = PHY has not detected any jabbering transmissions since PHYSTAT1 was last read

bit 0 **Unimplemented:** Read as '0'

Legend:			
R = Read-only bit	R/L = Read-only latch bit	U = Unimplemented bit, read as '0'	
'1' = Bit is set on POR	'0' = Bit is cleared on POR	LL = Bit latches low	LH = Bit latches high

REGISTER 3-7: PHSTAT2: PHYSICAL LAYER STATUS REGISTER 2

U-0	U-0	R-0	R-0	R-0	R-0	R-x	U-0
—	—	TXSTAT	RXSTAT	COLSTAT	LSTAT	DPXSTAT(1)	—
bit 15							bit 8

U-0	U-0	U-0	R-0	U-0	U-0	U-0	U-0
—	—	—	PLRITY	—	—	—	—
bit 7							bit 0

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **TXSTAT:** PHY Transmit Status bit
1 = PHY is transmitting data
0 = PHY is not transmitting data

bit 12 **RXSTAT:** PHY Receive Status bit
1 = PHY is receiving data
0 = PHY is not receiving data

bit 11 **COLSTAT:** PHY Collision Status bit
1 = A collision is occuring
0 = A collision is not occuring

bit 10 **LSTAT:** PHY Link Status bit (non-latching)
1 = Link is up
0 = Link is down

bit 9 **DPXSTAT:** PHY Duplex Status bit(1)
1 = PHY is configured for full-duplex operation (PHCON1.PDPXMD is set)
0 = PHY is configured for half-duplex operation (PHCON1.PDPXMD is clear)

Note 1: Reset values of the Duplex mode/status bits depend on the connection of the LED to the LEDB pin (see **Section 2.6 "LED Configuration"** for additional details).

bit 8-5 **Unimplemented:** Read as '0'

bit 4 **PLRITY:** Polarity Status bit
1 = The polarity of the signal on TPIN+/TPIN- is reversed
0 = The polarity of the signal on TPIN+/TPIN- is correct

bit 3-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

4.0 SERIAL PERIPHERAL INTERFACE (SPI)

4.1 Overview

The ENC28J60 is designed to interface directly with the Serial Peripheral Interface (SPI) port available on many microcontrollers. The implementation used on this device supports SPI mode 0,0 only. In addition, the SPI port requires that SCK be at Idle in a low state; selectable clock polarity is not supported.

Commands and data are sent to the device via the SI pin, with data being clocked in on the rising edge of SCK. Data is driven out by the ENC28J60 on the SO line, on the falling edge of SCK. The $\overline{CS}$ pin must be held low while any operation is performed and returned high when finished.

FIGURE 4-1: SPI™ INPUT TIMING

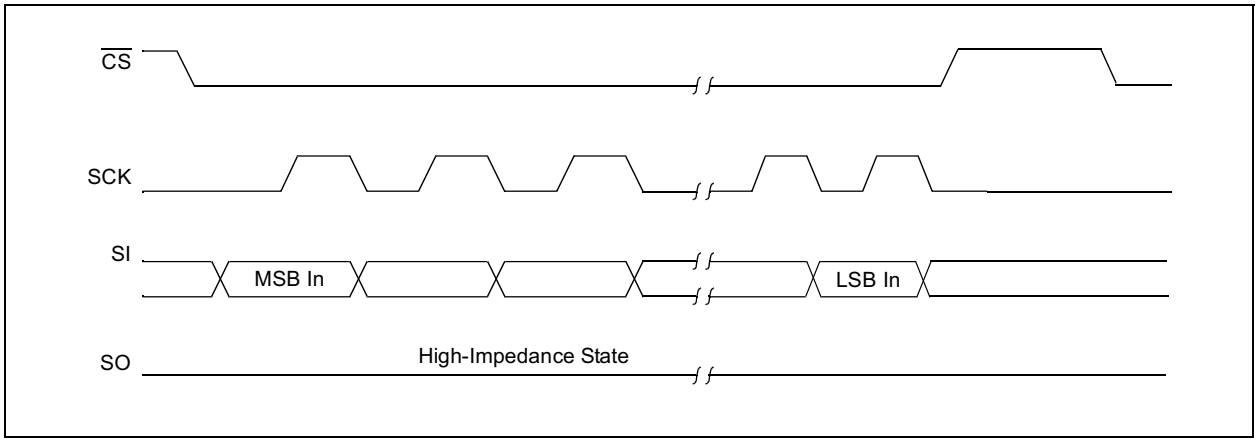
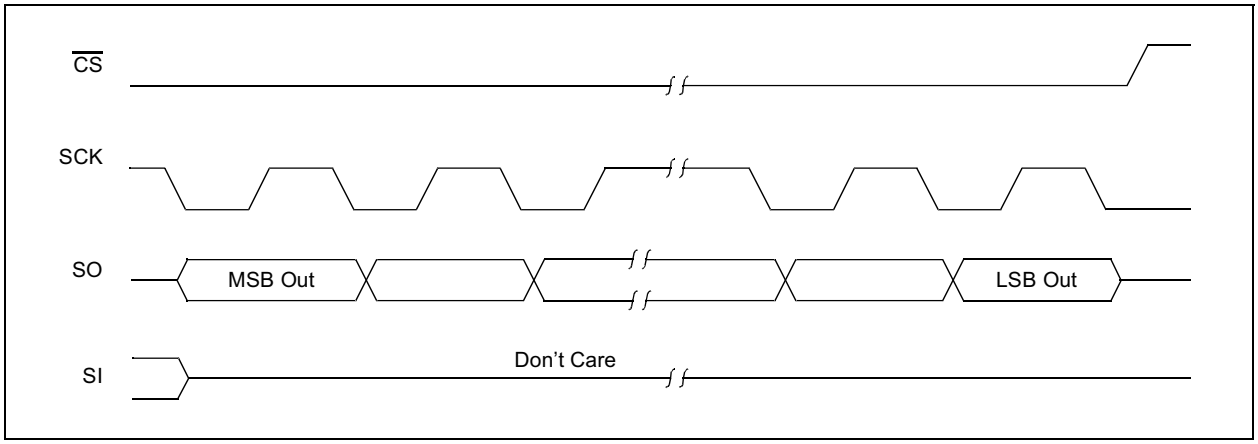


FIGURE 4-2: SPI™ OUTPUT TIMING



4.2 SPI Instruction Set

The operation of the ENC28J60 depends entirely on commands given by an external host controller over the SPI interface. These commands take the form of instructions, of one or more bytes, which are used to access the control memory and Ethernet buffer spaces. At the least, instructions consist of a 3-bit opcode, followed by a 5-bit argument that specifies either a register address or a data constant. Write and bit field instructions are also followed by one or more bytes of data.

A total of seven instructions are implemented on the ENC28J60. Table 4-1 shows the command codes for all operations.

TABLE 4-1: SPI™ INSTRUCTION SET FOR THE ENC28J60

Instruction Name and Mnemonic	Byte 0								Byte 1 and Following							
	Opcode			Argument					Data							
Read Control Register (RCR)	0	0	0	a	a	a	a	a	N/A							
Read Buffer Memory (RBM)	0	0	1	1	1	0	1	0	N/A							
Write Control Register (WCR)	0	1	0	a	a	a	a	a	d	d	d	d	d	d	d	d
Write Buffer Memory (WBM)	0	1	1	1	1	0	1	0	d	d	d	d	d	d	d	d
Bit Field Set (BFS)	1	0	0	a	a	a	a	a	d	d	d	d	d	d	d	d
Bit Field Clear (BFC)	1	0	1	a	a	a	a	a	d	d	d	d	d	d	d	d
System Command (Soft Reset) (SC)	1	1	1	1	1	1	1	1	N/A							

Legend: a = control register address, d = data payload.

4.2.1 READ CONTROL REGISTER COMMAND

The Read Control Register (RCR) command allows the host controller to read any of the ETH, MAC and MII registers in any order. The contents of the PHY registers are read via a special MII register interface (see **Section 3.3.1 "Reading PHY Registers"** for more information).

The RCR command is started by pulling the $\overline{CS}$ pin low. The RCR opcode is then sent to the ENC28J60, followed by a 5-bit register address (A4 through A0). The 5-bit address identifies any of the 32 control registers in the current bank. If the 5-bit address is an ETH register, then data in the selected register will immediately start shifting out MSb first on the SO pin. Figure 4-3 shows the read sequence for these registers.

If the address specifies one of the MAC or MII registers, a dummy byte will first be shifted out the SO pin. After the dummy byte, the data will be shifted out MSb first on the SO pin. The RCR operation is terminated by raising the $\overline{CS}$ pin. Figure 4-4 shows the read sequence for MAC and MII registers.

FIGURE 4-3: READ CONTROL REGISTER COMMAND SEQUENCE (ETH REGISTERS)

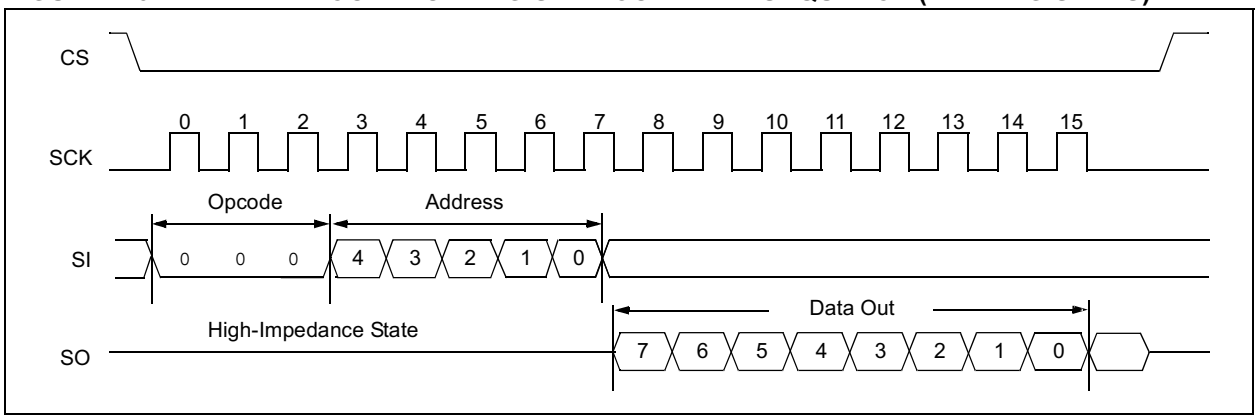
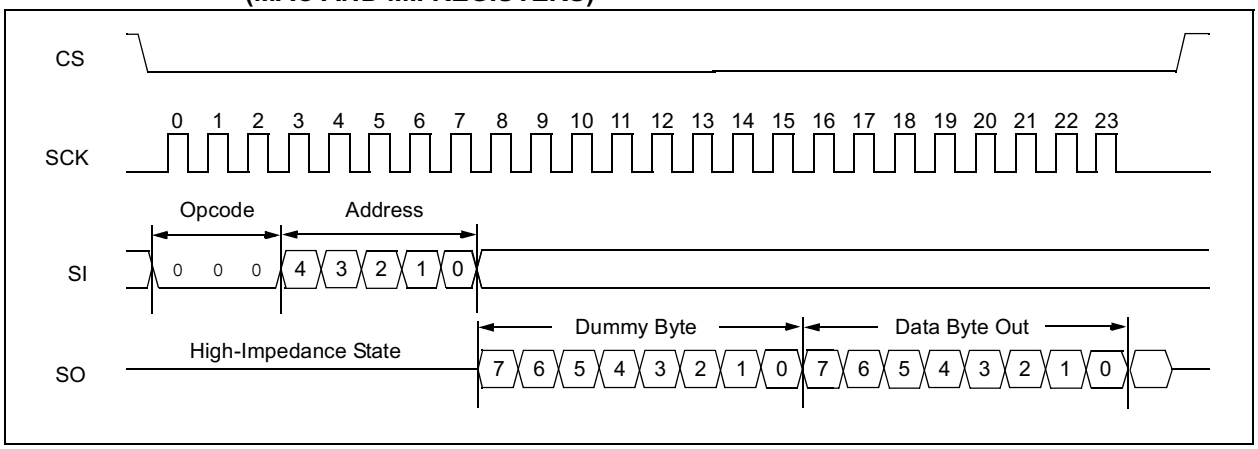


FIGURE 4-4: READ CONTROL REGISTER COMMAND SEQUENCE (MAC AND MII REGISTERS)



4.2.2 READ BUFFER MEMORY COMMAND

The Read Buffer Memory (RBM) command allows the host controller to read bytes from the integrated 8-Kbyte transmit and receive buffer memory.

If the AUTOINC bit in the ECON2 register is set, the ERDPT pointer will automatically increment to point to the next address after the last bit of each byte is read. The next address will normally be the current address incremented by one. However, if the last byte in the receive buffer is read (ERDPT = ERXND), the ERDPT pointer will change to the beginning of the receive buffer (ERXST). This allows the host controller to read packets from the receive buffer in a continuous stream without keeping track of when a wraparound is needed. If AUTOINC is set when address 1FFFh is read and ERXND does not point to this address, the read pointer will increment and wrap around to 0000h.

The RBM command is started by pulling the $\overline{CS}$ pin low. The RBM opcode is then sent to the ENC28J60, followed by the 5-bit constant 1Ah. After the RBM command and constant are sent, the data stored in the memory pointed to by ERDPT will be shifted out MSb first on the SO pin. If the host controller continues to provide clocks on the SCK pin, without raising $\overline{CS}$, the byte pointed to by ERDPT will again be shifted out MSb first on the SO pin. In this manner, with AUTOINC enabled, it is possible to continuously read sequential bytes from the buffer memory without any extra SPI command overhead. The RBM command is terminated by raising the $\overline{CS}$ pin.

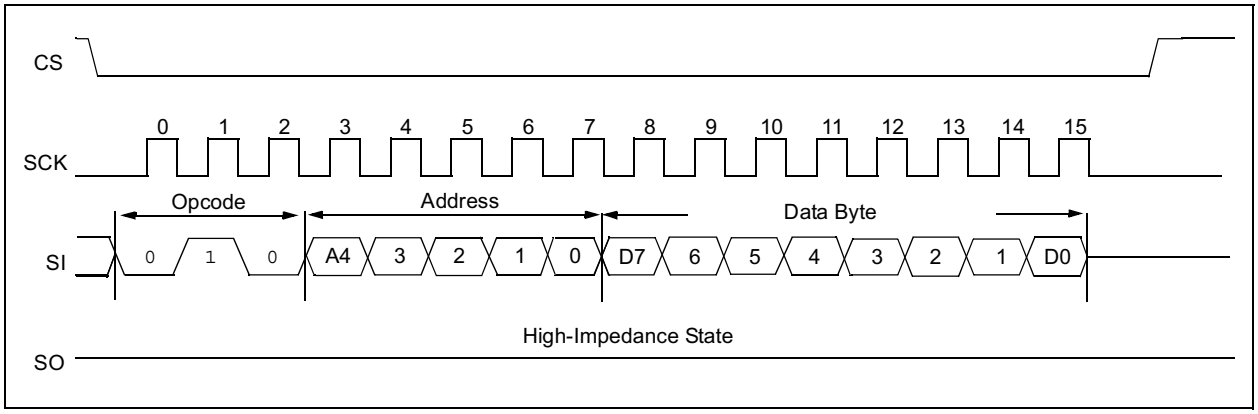
4.2.3 WRITE CONTROL REGISTER COMMAND

The Write Control Register (WCR) command allows the host controller to write to any of the ETH, MAC and MII Control registers in any order. The PHY registers are written to via a special MII register interface (see **Section 3.3.2 "Writing PHY Registers"** for more information).

The WCR command is started by pulling the $\overline{CS}$ pin low. The WCR opcode is then sent to the ENC28J60, followed by a 5-bit address (A4 through A0). The 5-bit address identifies any of the 32 control registers in the current bank. After the WCR command and address are sent, actual data that is to be written is sent, MSb first. The data will be written to the addressed register on the rising edge of the SCK line.

The WCR operation is terminated by raising the $\overline{CS}$ pin. If the $\overline{CS}$ line is allowed to go high before eight bits are loaded, the write will be aborted for that data byte. Refer to the timing diagram in Figure 4-5 for a more detailed illustration of the byte write sequence.

FIGURE 4-5: WRITE CONTROL REGISTER COMMAND SEQUENCE



4.2.4 WRITE BUFFER MEMORY COMMAND

The Write Buffer Memory (WBM) command allows the host controller to write bytes to the integrated 8-Kbyte transmit and receive buffer memory.

If the AUTOINC bit in the ECON2 register is set, after the last bit of each byte is written, the EWRPT pointer will automatically be incremented to point to the next sequential address (current address + 1). If address 1FFFh is written with AUTOINC set, the write pointer will increment to 0000h.

The WBM command is started by lowering the $\overline{CS}$ pin. The WBM opcode should then be sent to the ENC28J60, followed by the 5-bit constant 1Ah. After the WBM command and constant are sent, the data to be stored in the memory pointed to by EWRPT should be shifted out MSb first to the ENC28J60. After 8 data bits are received, the write pointer will automatically increment if AUTOINC is set. The host controller can continue to provide clocks on the SCK pin and send data on the SI pin, without raising $\overline{CS}$, to keep writing to the memory. In this manner, with AUTOINC enabled, it is possible to continuously write sequential bytes to the buffer memory without any extra SPI command overhead.

The WBM command is terminated by bringing up the $\overline{CS}$ pin. Refer to Figure 4-6 for a detailed illustration of the write sequence.

4.2.5 BIT FIELD SET COMMAND

The Bit Field Set (BFS) command is used to set up to 8 bits in any of the ETH Control registers. Note that this command cannot be used on the MAC registers, MII registers, PHY registers or buffer memory. The BFS command uses the provided data byte to perform a bit-wise OR operation on the addressed register contents.

The BFS command is started by pulling the $\overline{CS}$ pin low. The BFS opcode is then sent, followed by a 5-bit address (A4 through A0). The 5-bit address identifies any of the ETH registers in the current bank. After the BFS command and address are sent, the data byte containing the bit field set information should be sent, MSb first. The supplied data will be logically ORed to the content of the addressed register on the rising edge of the SCK line for the D0 bit.

If the $\overline{CS}$ line is brought high before eight bits are loaded, the operation will be aborted for that data byte. The BFS operation is terminated by raising the $\overline{CS}$ pin.

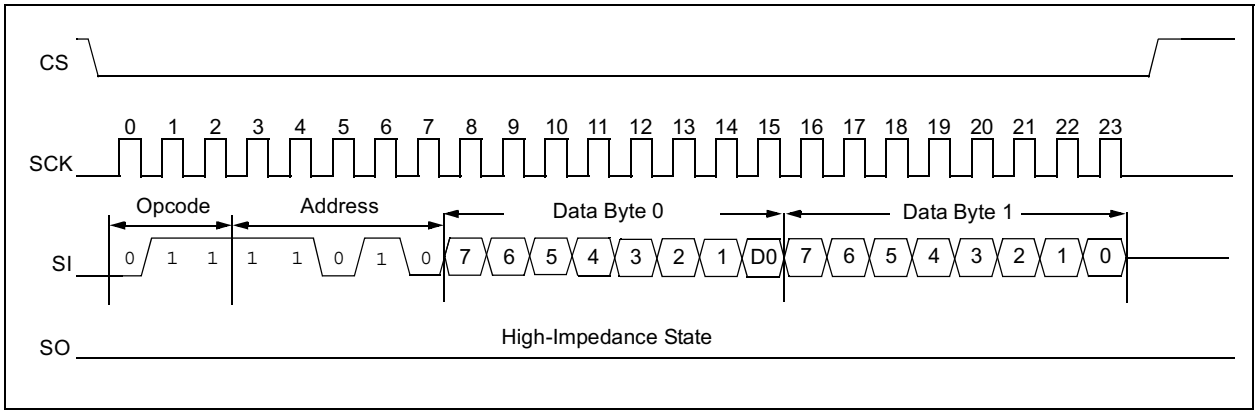
4.2.6 BIT FIELD CLEAR COMMAND

The Bit Field Clear (BFC) command is used to clear up to 8 bits in any of the ETH Control registers. Note that this command cannot be used on the MAC registers, MII registers, PHY registers or buffer memory. The BFC command uses the provided data byte to perform a bit-wise NOTAND operation on the addressed register contents. As an example, if a register had the contents of F1h and the BFC command was executed with an operand of 17h, then the register would be changed to have the contents of E0h.

The BFC command is started by lowering the $\overline{CS}$ pin. The BFC opcode should then be sent, followed by a 5-bit address (A4 through A0). The 5-bit address identifies any of the ETH registers in the current bank. After the BFC command and address are sent, a data byte containing the bit field clear information should be sent, MSb first. The supplied data will be logically inverted and subsequently ANDed to the contents of the addressed register on the rising edge of the SCK line for the D0 bit.

The BFC operation is terminated by bringing the $\overline{CS}$ pin high. If $\overline{CS}$ is brought high before eight bits are loaded, the operation will be aborted for that data byte.

FIGURE 4-6: WRITE BUFFER MEMORY COMMAND SEQUENCE



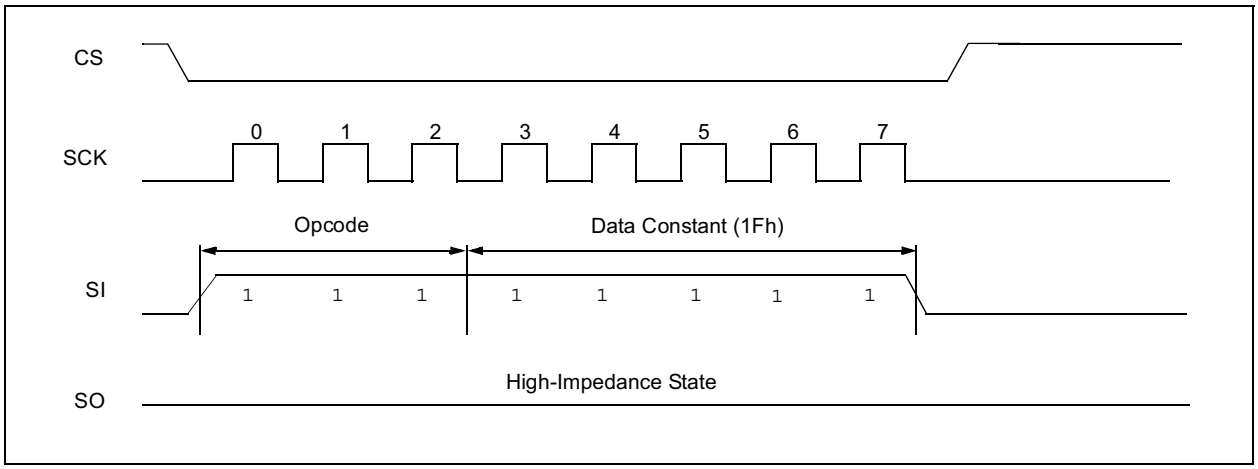
4.2.7 SYSTEM COMMAND

The System Command (SC) allows the host controller to issue a System Soft Reset command. Unlike other SPI commands, the SC is only a single-byte command and does not operate on any register.

The SC is started by pulling the $\overline{CS}$ pin low. The SC opcode should then be sent, followed by a 5-bit Soft Reset command constant of 1Fh. The SC operation is terminated by raising the $\overline{CS}$ pin. Figure 4-7 shows a detailed illustration of the System Command sequence.

For more information on SC's Soft Reset, refer to **Section 11.2 "System Reset"**.

FIGURE 4-7: SYSTEM COMMAND SEQUENCE



5.0 ETHERNET OVERVIEW

Before discussing the use of the ENC28J60 as an Ethernet interface, it may be helpful to review the structure of a typical data frame. Users requiring more information should refer to IEEE Standard 802.3 which is the basis for the Ethernet protocol.

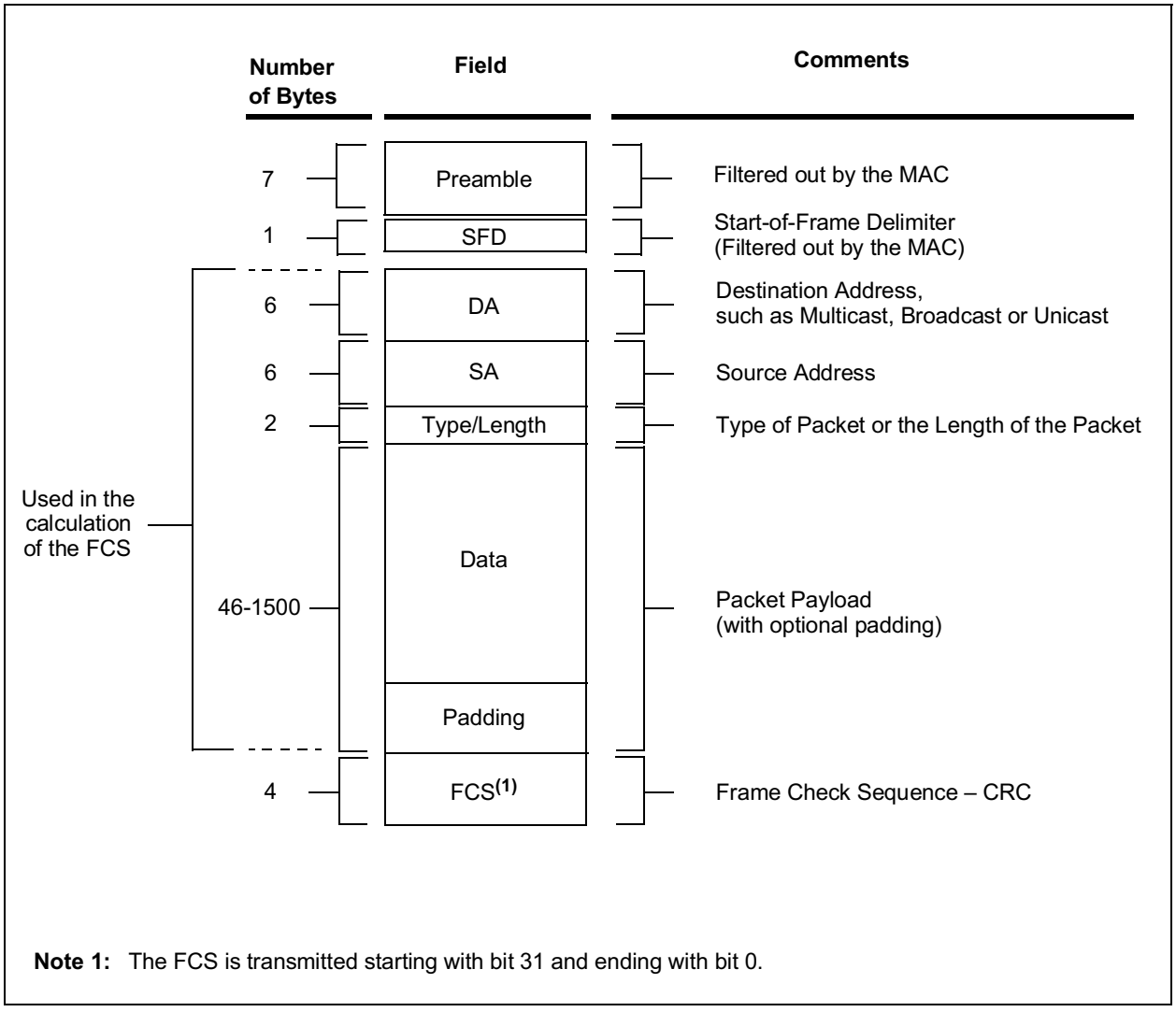
5.1 Packet Format

Normal IEEE 802.3 compliant Ethernet frames are between 64 and 1518 bytes long. They are made up of five or six different fields: a destination MAC address, a source MAC address, a type/length field, data payload, an optional padding field and a Cyclic Redundancy Check (CRC). Additionally, when transmitted on the Ethernet medium, a 7-byte preamble field and start-of-frame delimiter byte are appended to the beginning of the Ethernet packet. Thus, traffic seen on the twisted pair cabling will appear as shown in Figure 5-1.

5.1.1 PREAMBLE/START-OF-FRAME DELIMITER

When transmitting and receiving data with the ENC28J60, the preamble and start of frame delimiter bytes will automatically be generated or stripped from the packets when they are transmitted or received. The host controller does not need to concern itself with them. Normally, the host controller will also not need to concern itself with padding and the CRC which the ENC28J60 will also be able to automatically generate when transmitting and verify when receiving. The padding and CRC fields will, however, be written into the receive buffer when packets arrive, so they may be evaluated by the host controller if needed.

FIGURE 5-1: ETHERNET PACKET FORMAT



Note 1: The FCS is transmitted starting with bit 31 and ending with bit 0.

5.1.2 DESTINATION ADDRESS

The destination address field is a 6-byte field filled with the MAC address of the device that the packet is directed to. If the Least Significant bit in the first byte of the MAC address is set, the address is a multicast destination. For example, 01-00-00-00-F0-00 and 33-45-67-89-AB-CD are multicast addresses, while 00-00-00-00-F0-00 and 32-45-67-89-AB-CD are not.

Packets with multicast destination addresses are designed to arrive and be important to a selected group of Ethernet nodes. If the destination address field is the reserved multicast address, FF-FF-FF-FF-FF-FF, the packet is a broadcast packet and it will be directed to everyone sharing the network. If the Least Significant bit in the first byte of the MAC address is clear, the address is a unicast address and will be designed for usage by only the addressed node.

The ENC28J60 incorporates receive filters which can be used to discard or accept packets with multicast, broadcast and/or unicast destination addresses. When transmitting packets, the host controller is responsible for writing the desired destination address into the transmit buffer.

5.1.3 SOURCE ADDRESS

The source address field is a 6-byte field filled with the MAC address of the node which created the Ethernet packet. Users of the ENC28J60 must generate a unique MAC address for each controller used.

MAC addresses consist of two portions. The first three bytes are known as the Organizationally Unique Identifier (OUI). OUIs are distributed by the IEEE. The last three bytes are address bytes at the discretion of the company that purchased the OUI.

When transmitting packets, the assigned source MAC address must be written into the transmit buffer by the host controller. The ENC28J60 will not automatically transmit the contents of the MAADR registers which are used for the unicast receive filter and unicast WOL filter.

5.1.4 TYPE/LENGTH

The type/length field is a 2-byte field which defines which protocol the following packet data belongs to. Alternately, if the field is filled with the contents of 05DCh (1500) or any smaller number, the field is considered a length field and it specifies the amount of non-padding data which follows in the data field. Users implementing proprietary networks may choose to treat this field as a length field, while applications implementing protocols such as the Internet Protocol (IP) or Address Resolution Protocol (ARP), should program this field with the appropriate type defined by the protocol's specification when transmitting packets.

5.1.5 DATA

The data field is a variable length field anywhere from 0 to 1500 bytes. Larger data packets will violate Ethernet standards and will be dropped by most Ethernet nodes. The ENC28J60, however, is capable of transmitting and receiving larger packets when the Huge Frame Enable bit is set (MACON3.HFRMEN = 1).

5.1.6 PADDING

The padding field is a variable length field added to meet IEEE 802.3 specification requirements when small data payloads are used. The destination, source, type, data and padding of an Ethernet packet must be no smaller than 60 bytes. Adding the required 4-byte CRC field, packets must be no smaller than 64 bytes. If the data field is less than 46 bytes long, a padding field is required.

When transmitting packets, the ENC28J60 automatically generates zero padding if the MACON3.PADCFG<2:0> bits are configured to do so. Otherwise, the host controller should manually add padding to the packet before transmitting it. The ENC28J60 will not prevent the transmission of undersize packets should the host controller command such an action.

When receiving packets, the ENC28J60 automatically rejects packets which are less than 18 bytes. All packets 18 bytes and larger will be subject to the standard receive filtering criteria and may be accepted as normal traffic.

5.1.7 CRC

The CRC field is a 4-byte field which contains an industry standard 32-bit CRC calculated with the data from the destination, source, type, data and padding fields.

When receiving packets, the ENC28J60 will check the CRC of each incoming packet. If ERXFCON.CRCEN is set, packets with invalid CRCs will automatically be discarded. If CRCEN is clear and the packet meets all other receive filtering criteria, the packet will be written into the receive buffer and the host controller will be able to determine if the CRC was valid by reading the receive status vector (see **Section 7.2 "Receiving Packets"**).

When transmitting packets, the ENC28J60 will automatically generate a valid CRC and transmit it if the MACON3.PADCFG<2:0> bits are configured to cause this. Otherwise, the host controller must generate the CRC and place it in the transmit buffer. Given the complexity of calculating a CRC, it is highly recommended that the PADCFG bits be configured such that the ENC28J60 will automatically generate the CRC field.

