International Journal of Advanced Research in Engineering and Applied Sciences

ISSN: 2278-6252

ADVANCED ARTIFICIAL EYE IMPLEMENTATION FOR DETECTING MONOCHROME OBJECTS FOR INDUSTRIAL AUTOMATION USING ARM7 TDMI BASED LPC2148 MICROCONTROLLER

V. Mallikarjuna*

B. Rama Bhupal Reddy**

Abstract: *This Paper is the comprehensive review of implementation of advanced artificial eye implementation for detecting monochrome objects for industrial automation. In many situations, autonomous robots can provide effective solution to menial or dangerous tasks. In this case, it is desirable to create an autonomous robot that can identify objects from conveyor belts and relocate them if the object meet certain criteria. Obviously, when dealing with a large number of objects, this is a very menial task. A geared DC motor with 60RPM is used to control the conveyor belt. Robotic free balance wheel is used to support the conveyor belt on the other end. Here we will be using picking arm using controller motor to pick the particular object from the belt and place it according to the colour sensing. This paper uses low power supplies, one is regulated 5V modules and other one is 3.3V for LPC2148. 7805 three terminal voltage regulator is used for voltage regulation. Bridge type full wave rectifier is used to rectify the ac output of secondary of 230/12V step down transformer.*

Keywords: *Embedded system Design, +9V, 500mA Regulated Power Supply, LPC2148 controller.*

*Assistant Professor in Mechanical Department, Sri Venkateswara Institute of Science and Technology, Kadapa, A.P. - India

**Associate Professor in Mathematics, K.S.R.M. College of Engineering, Kadapa, A.P. - India

1. INTRODUCTION

This robot is used for Controlling the speed and direction by using Zigbee Technology. The concept is we can control direction of robot with sensing of gas leakage in industries.

The paper is built around ARM Technology, in which we are using LPC2148 controller is based on a 16/32 bit ARM7TDMI-S™ CPU. By using GPIO pins of the controller we can receive the signals getting from the colour sensor and thereby controlling the motor direction and speed by using H-Bridge (L293D). If it senses any gas leakage automatically a buzzer will be switched ON. And this information will be wirelessly transmitted to receiver end using X bee communication. Many industrial object-sorting applications leverage benefits of hyper spectral imaging technology. Design of object sorting algorithms is a challenging pattern recognition problem due to its multi-level nature. Objects represented by sets of pixels/spectra in hyper spectral images are to be allocated into pre-specified sorting categories. Sorting categories are often defined in terms of lower-level concepts such as material or defect types. This paper illustrates the design of two-stage sorting algorithms, learning to discriminate individual pixels/spectra and fusing the per-pixel decisions into a single per-object outcome. The paper provides a case-study on algorithm design in a real-world industrial sorting problem. Four groups of algorithms are studied varying the level of prior knowledge about the sorting problem. Apart of the sorting accuracy, the algorithm execution speed is estimated assuming an ideal implementation. Relating these two performance criteria allows us to discuss the accuracy/speed trade-off of different algorithms.

2. OBJECT SORTING SYSTEM

The object sorting system based on spectral imaging receives an image stream on its input. Objects, present in the stream, are detected and each of them is classified into one of the pre-specified sorting categories. In this study, we assume a perfect object detector and focus entirely on the design of the object classifier. Surveying the real-world object sorting problems we can observe that the sorting categories are usually defined using lower-level concepts For example, in potato sorting the high-level category defected potato is described by fractions (percentiles) of damage, rot, or greening present in the object. In this study, we follow the approach where the classifiers of the high level classes or lower-level concepts are trained based on individual pixels/spectra within the objects. The object

decision is then performed by the fusion of lower-level decisions. From a pattern recognition viewpoint, such sorting system exhibits multiple levels, namely pixels/spectra, objects, lower-level concepts modes), and high-level categories (classes).

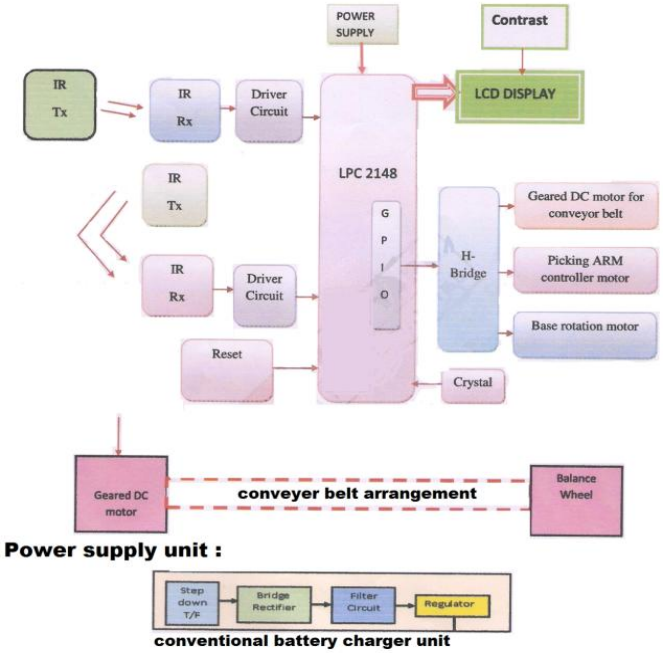


Figure 1: Block diagram

Infrared

Infrared is a energy radiation with a frequency below our eyes sensitivity, so we cannot see it. Even that we cannot "see" sound frequencies, we know that it exist, we can listen them.

Figure 2: Sound frequencies

Even that we cannot see or hear infrared, we can feel it at our skin temperature sensors. When you approach your hand to fire or warm element, you will “feel” the heat, but you can't see it. You can see the fire because it emits other types of radiation, visible to your eyes, but it also emits lots of infrared that you can only feel in your skin.

Infrared in Electronics

Infra-Red is interesting, because it is easily generated and doesn't suffer electromagnetic interference, so it is nicely used to communication and control, but it is not perfect, some

other light emissions could contains infrared as well, and that can interfere in this communication. The sun is an example, since it emits a wide spectrum or radiation.

The adventure of using lots of infra-red in TV/VCR remote controls and other applications, brought infra-red diodes (emitter and receivers) at very low cost at the market. From now on you should think as infrared as just a “red” light. This light can means something to the receiver, the “on or off” radiation can transmit different meanings. Lots of things can generate infrared, anything that radiate heat do it, including out body, lamps, stove, oven, friction your hands together, even the hot water at the faucet.

To allow a good communication using infra-red, and avoid those “fake” signals, it is imperative to use a “key” that can tell the receiver what is the real data transmitted and what is fake. As an analogy, looking eye naked to the night sky you can see hundreds of stars, but you can spot easily a far away airplane just by its flashing strobe light. That strobe light is the “key”, the "coding" element that alerts us.

Similar to the airplane at the night sky, our TV room may have hundreds of tinny IR sources, our body, and the lamps around, even the hot cup of tea. A way to avoid all those other sources, is generating a key, like the flashing airplane. So, remote controls use to pulsate its infrared in a certain frequency. The IR receiver module at the TV, VCR or stereo “tunes” to this certain frequency and ignores all other IR received. The best frequency for the job is between 30 and 60 kHz, the most used is around 36 kHz.

IR Generation

To generate a 36 kHz pulsating infrared is quite easy, more difficult is to receive and identify this frequency. This is why some companies produce infrared receives, that contains the filters, decoding circuits and the output shaper, that delivers a square wave, meaning the existence or not of the 36kHz incoming pulsating infrared. It means that those 3 dollars small units, have an output pin that goes high (+5V) when there is a pulsating 36kHz infrared in front of it, and zero volts when there is not this radiation.

A square wave of approximately 27uS (microseconds) injected at the base of a transistor, can drive an infrared LED to transmit this pulsating light wave. Upon its presence, the commercial receiver will switch its output to high level (+5V).If you can turn on and off this frequency at the transmitter; your receiver's output will indicate when the transmitter is on

or off. Those IR demodulators have inverted logic at its output, when a burst of IR is sensed it drives its output to low level, meaning logic level = 1.

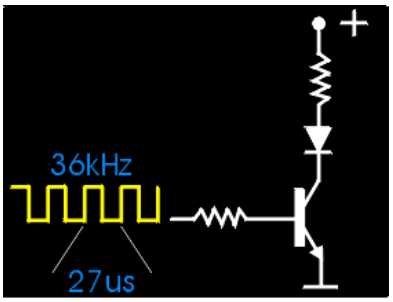


Figure 3: Transmitter frequency

The TV, VCR, and Audio equipment manufacturers for long use infra-red at their remote controls. To avoid a Philips remote control to change channels in a Panasonic TV, they use different codification at the infrared, even that all of them use basically the same transmitted frequency, from 36 to 50 kHz. So, all of them use a different combination of bits or how to code the transmitted data to avoid interference.

RC-5

Various remote control systems are used in electronic equipment today. The RC5 control protocol is one of the most popular and is widely used to control numerous home appliances, entertainment systems and some industrial applications including utility consumption remote meter reading, contact-less apparatus control, telemetry data transmission, and car security systems. Philips originally invented this protocol and virtually all Philips' remotes use this protocol. Following is a description of the RC5. When the user pushes a button on the hand-held remote, the device is activated and sends modulated infrared light to transmit the command. The remote separates command data into packets. Each data packet consists of a 14-bit data word, which is repeated if the user continues to push the remote button. The data packet structure is as follows:

- 2 start bits
- 1 control bit
- 5 address bits
- 6 command bits.

The start bits are always logic '1' and intended to calibrate the optical receiver automatic gain control loop. Next, is the control bit. This bit is inverted each time the user releases the remote button and is intended to differentiate situations when the user continues to hold

the same button or presses it again. The next 5 bits are the address bits and select the destination device. A number of devices can use RC5 at the same time. To exclude possible interference, each must use a different address. The 6 command bits describe the actual command. As a result, a RC5 transmitter can send the 2048 unique commands. The transmitter shifts the data word, applies Manchester encoding and passes the created one-bit sequence to a control carrier frequency signal amplitude modulator. The amplitude modulated carrier signal is sent to the optical transmitter, which radiates the infrared light. In RC5 systems the carrier frequency has been set to 36 kHz. Figure below displays the RC5 protocol.

The receiver performs the reverse function. The photo detector converts optical transmission into electric signals, filters it and executes amplitude demodulation. The receiver output bit stream can be used to decode the RC5 data word. This operation is done by the microprocessor typically, but complete hardware implementations are present on the market as well. Single-die optical receivers are being mass produced by a number of companies such as Siemens, Temic, Sharp, Xiamen Hualian, Japanese Electric and others. Please note that the receiver output is inverted (log. 1 corresponds to illumination absence).

IR Receiver

The TSOP17 – series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.

The demodulated output signal can directly be decoded by a microprocessor. TSOP17 is the standard IR remote control receiver series, supporting all major transmission codes.

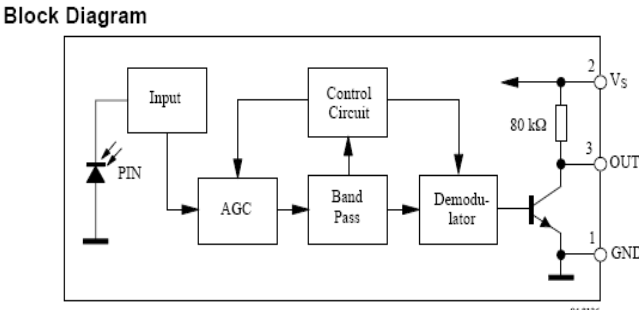


Figure 4 : IR Receiver

ISSN: 2278-6252

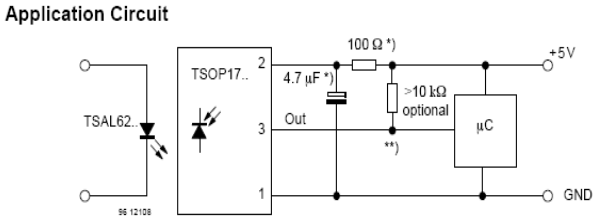


Fig.5: Application Circuit for IR Receiver

Suitable Data Format

The circuit of the TSOP17 is designed in that way that unexpected output pulses due to noise or disturbance signals are avoided. A band pass filter, an integrator stage and an automatic gain control are used to suppress such disturbances. The distinguishing mark between data signal and disturbance signal are carrier frequency, burst length and duty cycle. The data signal should fulfil the following condition

- Carrier frequency should be close to centre frequency of the band pass (e.g. 38 kHz).
- Burst length should be 10cycles/burst or longer.
- After each burst which is between 10 cycles and 70 cycles a gap time of at least 14 cycles is necessary for each burst which is longer than 1.8ms a corresponding gap time is necessary at some time in the data stream. This gap time should have at least same length as the burst.
- Up to 1400 short bursts per second can be received continuously.

Some examples for suitable data format are: NEC Code, Toshiba Micom Format, Sharp Code, RC5 Code, RC6 Code, R–2000 Code, Sony Format (SIRCS). When a disturbance signal is applied to the TSOP17.. it can still receive the data signal. However the sensitivity is reduced to that level that no unexpected pulses will occur. Some examples for such disturbance signals which are suppressed by the TSOP17 are

- DC light (e.g. from tungsten bulb or sunlight)
- Continuous signal at 38 kHz or at any other frequency

Signals from fluorescent lamps with electronic ballast (an example of the signal modulation is in the figure below)

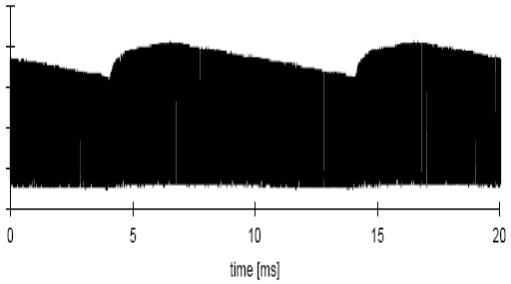


Figure 6: IR Signal Fluorescent Lamp with Low Modulation

IR Section - IR Tx.:

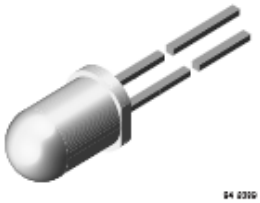


Figure 7: IR Sensor

TSAL6200 is a high efficiency infrared emitting diode in GaAlAs on GaAs technology, molded in clear, bluegrey tinted plastic packages. In comparison with the standard GaAs on GaAs technology these emitters achieve more than 100% radiant power improvement at a similar wavelength. The forward voltages at low current and at high pulse current roughly correspond to the low values of the standard technology. Therefore these emitters are ideally suitable as high performance replacements of standard emitters.

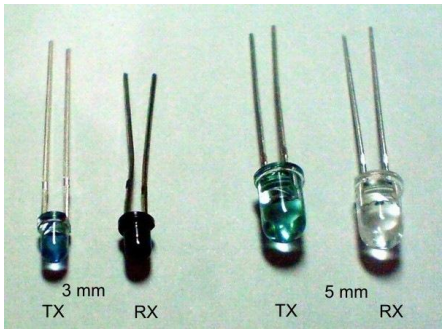


Figure 8: IR Transmitter & IR Receiver

IR Rx:

Figure 9: Transistor

The TSOP17 – series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.

The demodulated output signal can directly be decoded by a microprocessor. TSOP17XX is the standard IR remote control receiver series, supporting all major transmission codes.

Features

- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against electrical field disturbance
- TTL and CMOS compatibility
- Output active low
- Low power consumption
- High immunity against ambient light
- Continuous data transmission possible (up to 2400 bps)
- Suitable burst length 10cycles/burst.

3. ARM PROCESSOR

Founded in November 1990, it is spun out of Acorn Computers, it Designs the ARM range of RISC processor cores. Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers. ARM does not fabricate silicon itself, it also develop technologies to assist with the design-in of the ARM architecture. Software tools, boards, debug hardware, application software, bus architectures, peripherals etc.

The ARM processor core originates within a British computer company called Acorn. In the mid-1980s they were looking for replacement for the 6502 processor used in their BBC computer range, which were widely used in UK schools. None of the 16-bit architectures becoming available at that time met their requirements, so they designed their own 32-bit processor.

Other companies became interested in this processor, including Apple who was looking for a processor for their PDA project (which became the Newton). After much discussion this led to Acorn's processor design team splitting off from Acorn at the end of 1990 to become Advanced RISC Machines Ltd, now just ARM Ltd. Thus ARM Ltd now designs the ARM family of RISC processor cores, together with a range of other supporting technologies.

One important point about ARM is that it does not fabricate silicon itself, but instead just produces the design - we are an Intellectual Property (or IP) company. Instead silicon is produced by companies who license the ARM processor design.

Architectural overview

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core. Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue. The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM7TDMI-S processor has two instruction sets

- The standard 32-bit ARM set.
- A 16-bit Thumb set.

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65 % of the code size of ARM, and 160 % of the performance of an equivalent ARM processor connected to a 16-bit memory system. The particular flash implementation in the LPC2141/42/44/46/48 allows for full speed execution also in ARM mode. It is recommended to program performance critical and short code sections (such as interrupt service routines and DSP algorithms) in ARM mode. The impact on the overall code size will be minimal but the speed can be increased by 30% over Thumb mode.

ARM Powered Products in Industry

Data Sizes and Instruction Sets

- The ARM is a 32-bit architecture.
- When used in relation to the ARM:
  - Byte means 8 bits
  - Half word means 16 bits (two bytes)
  - Word means 32 bits (four bytes)
  - Most ARM's implement two instruction sets
  - 32-bit ARM Instruction Set
  - 16-bit Thumb Instruction Set
- Jazelle cores can also execute Java byte code

ARM7TDMI Processor Core

- Current low-end ARM core for applications like digital mobile phones
- TDMI
  - T: Thumb, 16-bit compressed instruction set
  - D: on-chip Debug support, enabling the processor to halt in response to a debug request
  - M: enhanced Multiplier, yield a full 64-bit result, high performance
  - I: Embedded ICE hardware
- Von Neumann architecture

ARM7TDMI Core Diagram

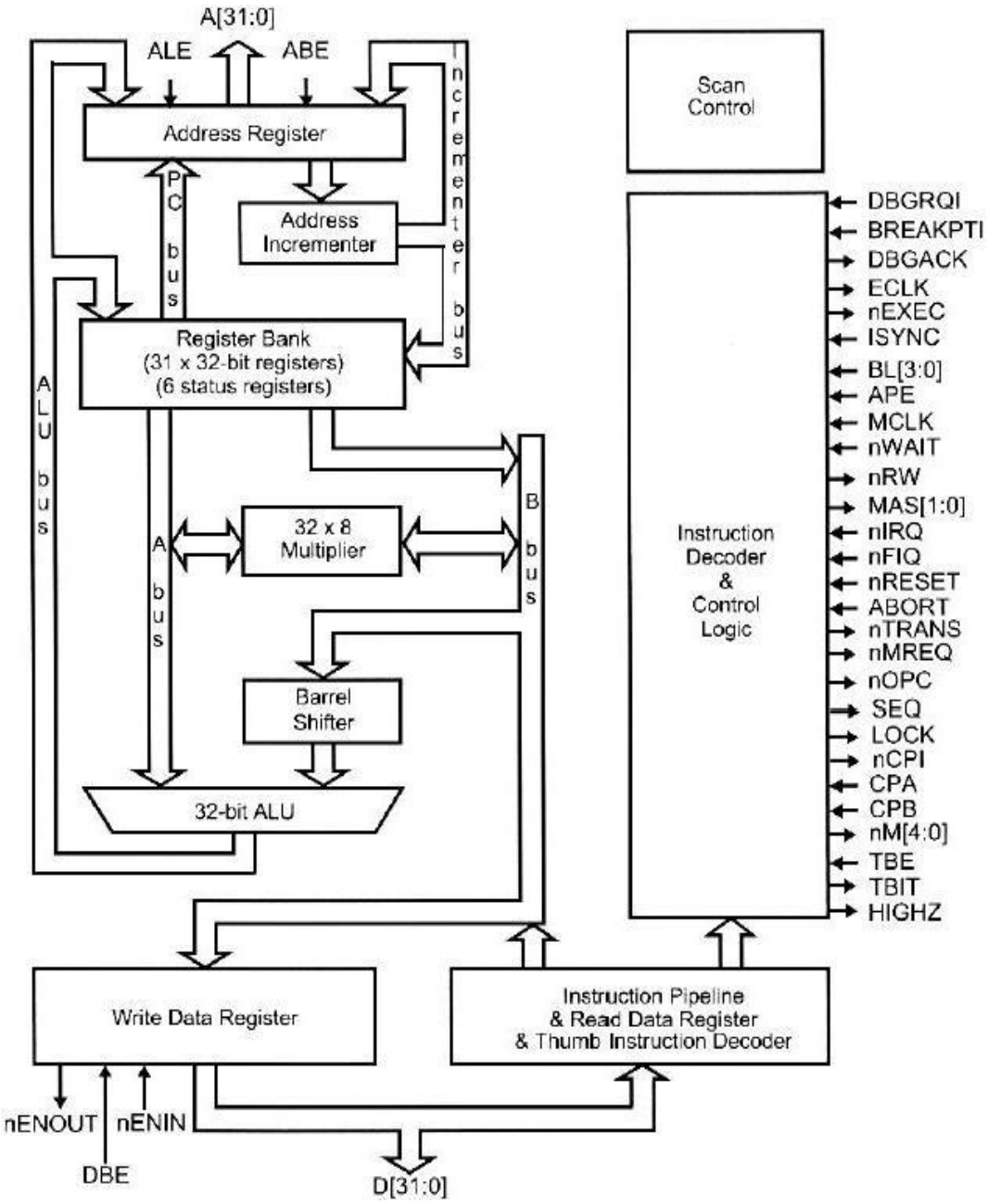


Figure 10: ARM7TDMI Core Diagram

ARM7TDMI Interface Signals

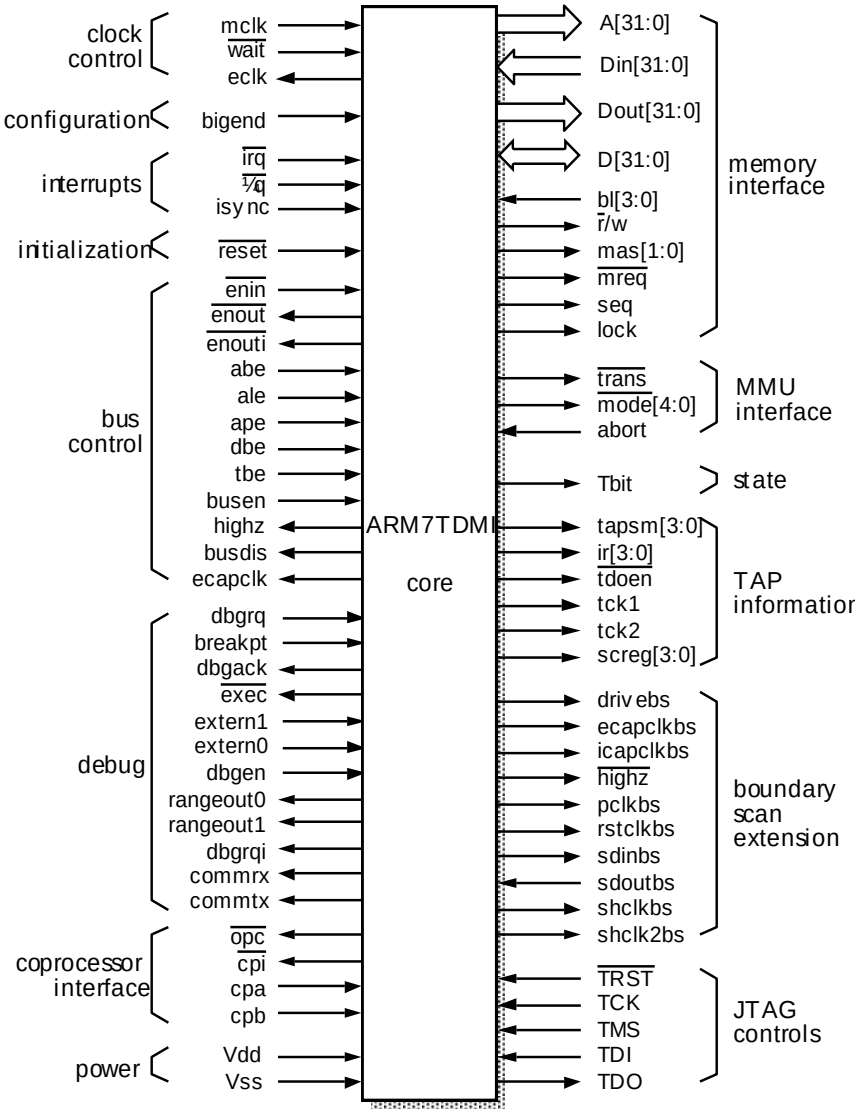


Figure 11: ARM7TDMI Interface Signals

- Clock control
  - All state change within the processor are controlled by mclk, the memory clock
  - Internal clock = mclk AND \wait
  - eclk clock output reflects the clock used by the core
- Memory interface
  - 32-bit address A[31:0], bidirectional data bus D[31:0], separate data out Dout[31:0], data in Din[31:0]
  - seq indicates that the memory address will be sequential to that used in the previous cycle

- Interrupt
  - \fiq, fast interrupt request, higher priority
  - \irq, normal interrupt request
  - isync, allow the interrupt synchronizer to be passed
- Initialization
  - \reset, starts the processor from a known state, executing from address

ARM7TDMI characteristics

Process	0.35um	Transistors	74209	MIPS	60
Metal layers	3	Core area	2.1mm^2	Power	87 mW
Vdd	3.3V	Clock	0 to 66 MHz	MIP S/W	690

Processor Core Vs CPU Core

- Processor Core
  - The engine that fetches instructions and execute them
  - E.g.: ARM7TDMI, ARM9TDMI, ARM9E-S
- CPU Core
  - Consists of the ARM processor core and some tightly coupled function blocks
    - Cache and memory management blocks
    - E.g.: ARM710T, ARM720T, ARM74T, ARM920T, ARM922T, ARM940T, ARM946E-S, and ARM966E-S

4. LPC2148 CONTROLLER

The LPC2141/42/44/46/48 microcontrollers are based on a 16-bit/32-bit ARM7TDMI-CPU with real-time emulation and embedded trace support, that combine microcontroller with embedded high speed flash memory ranging from 32kB to 512kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty. Due to their tiny size and low power consumption, LPC2141/42/44/46/48 are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. Serial communications interfaces ranging from a USB 2.0 Full-speed device, multiple UARTs, SPI, SSP to I2C-bus and on-chip SRAM of 8kB up to 40kB, make these devices very well suited for communication

gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWM channels and 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers suitable for industrial control and medical systems.

Key Features

- 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 8kB to 40kB of on-chip static RAM and 32kB to 512kB of on-chip flash memory.
- 128-bit wide interface/accelerator enables high-speed 60 MHz operation
- In-System Programming / In-Application Programming (ISP/IAP) via on-chip boot loader Software. Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1 ms.
- Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high-speed tracing of instruction execution.
- USB 2.0 Full-speed compliant device controller with 2kB of endpoint RAM.
- In addition, the LPC2146/48 provides 8kB of on-chip RAM accessible to USB by DMA.
- One or two (LPC2141/42 vs. LPC2144/46/48) 10-bit ADCs provide a total of 6/14
- Analog inputs, with conversion times as low as 2.44µs per channel.
- Single 10-bit DAC provides variable analog output (LPC2142/44/46/48 only).
- Two 32-bit timers/external event counters (with four capture and four compare Channels each), PWM unit (six outputs) and watchdog.
- Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input
- Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400kbit/s), SPI and SSP with buffering and variable data length capabilities.
- Vectored Interrupt Controller (VIC) with configurable priorities and vector addresses.
- Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.
- Up to 21 external interrupt pins available.
- 60 MHz maximum CPU clock available from programmable on-chip PLL with settling time of 100µs.
- On-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz.

- Power saving modes include Idle and Power-down.
- Individual enable/disable of peripheral functions as well as peripheral clock scaling for additional power optimization.
- Processor wake-up from Power-down mode via external interrupt or BOD.
- Single power supply chip with POR and BOD circuits:
- CPU operating voltage range of 3.0 V to 3.6 V (3.3 V ± 10 %) with 5 V tolerant I/O

Ordering information:

Table 1: Ordering information

Type number	Package		
	Name	Description	Version
LPC2141FBD64	LQFP64	plastic low profile quad flat package; 64 leads; body 10 × 10 × 1.4 mm	SOT314-2
LPC2142FBD64			
LPC2144FBD64			
LPC2146FBD64			
LPC2148FBD64			

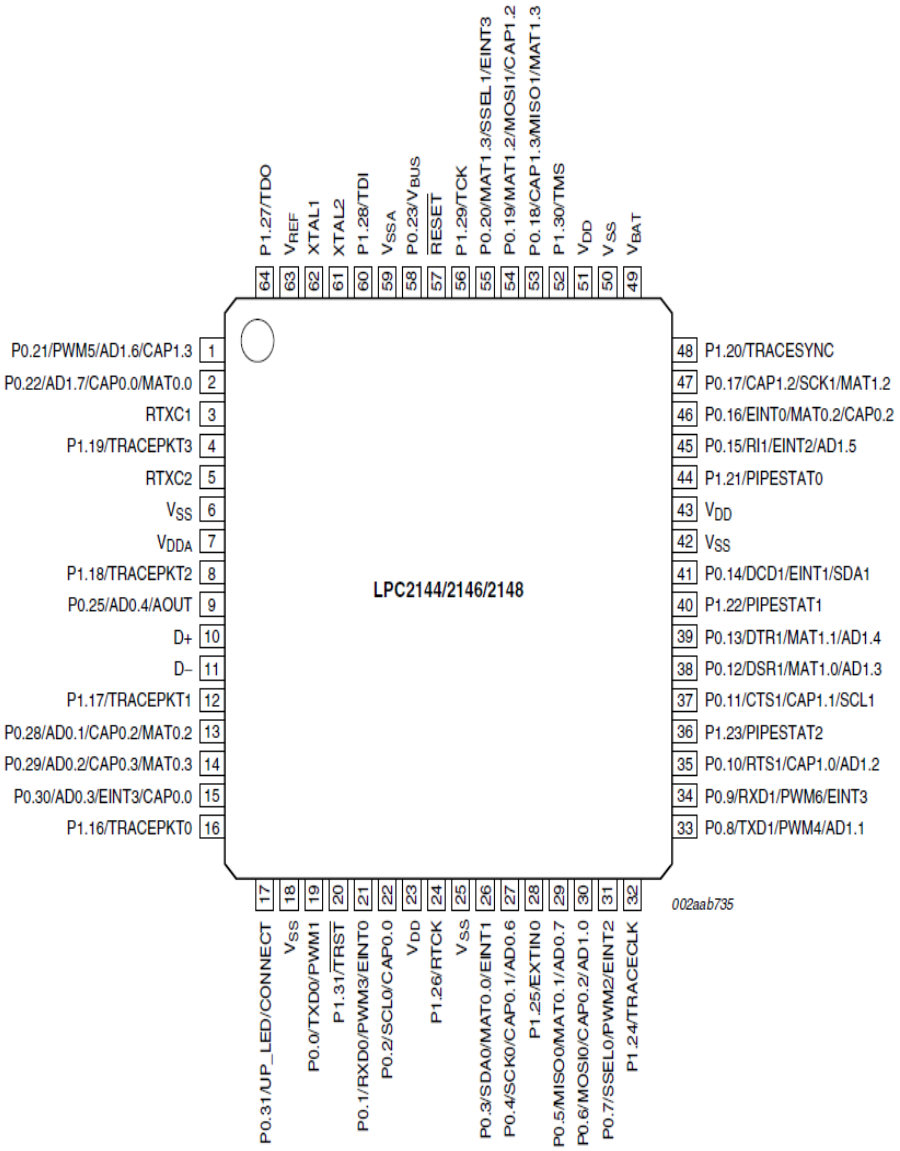
Ordering Options

Table 2: Ordering options

Type number	Flash memory	RAM	Endpoint USB RAM	ADC (channels overall)	DAC	Temperature range (°C)
LPC2141FBD64	32 kB	8 kB	2 kB	1 (6 channels)	-	–40 to +85
LPC2142FBD64	64 kB	16 kB	2 kB	1 (6 channels)	1	–40 to +85
LPC2144FBD64	128 kB	16 kB	2 kB	2 (14 channels)	1	–40 to +85
LPC2146FBD64	256 kB	32 kB + 8 kB shared with USB DMA [1]	2 kB	2 (14 channels)	1	–40 to +85
LPC2148FBD64	512 kB	32 kB + 8 kB shared with USB DMA [1]	2 kB	2 (14 channels)	1	–40 to +85

ISSN: 2278-6252

Pin Diagram



Port Pin Description

Symbol	Pin	Type	Description
P0.9/RXD1/ PWM6/EINT3	34 [2]	I/O	**P0.9** — General purpose input/output digital pin (GPIO).
		I	**RXD1** — Receiver input for UART1.
		O	**PWM6** — Pulse Width Modulator output 6.
		I	**EINT3** — External interrupt 3 input.
P0.10/RTS1/ CAP1.0/AD1.2	35 [4]	I/O	**P0.10** — General purpose input/output digital pin (GPIO).
		O	**RTS1** — Request to Send output for UART1. LPC2144/46/48 only.
		I	**CAP1.0** — Capture input for Timer 1, channel 0.
		I	**AD1.2** — ADC 1, input 2. Available in LPC2144/46/48 only.
P0.11/CTS1/ CAP1.1/SCL1	37 [3]	I/O	**P0.11** — General purpose input/output digital pin (GPIO).
		I	**CTS1** — Clear to Send input for UART1. Available in LPC2144/46/48 only.
		I	**CAP1.1** — Capture input for Timer 1, channel 1.
		I/O	**SCL1** — I^2C1 clock input/output. Open-drain output (for I^2C-bus compliance)
P0.12/DSR1/ MAT1.0/AD1.3	38 [4]	I/O	**P0.12** — General purpose input/output digital pin (GPIO).
		I	**DSR1** — Data Set Ready input for UART1. Available in LPC2144/46/48 only.
		O	**MAT1.0** — Match output for Timer 1, channel 0.
		I	**AD1.3** — ADC input 3. Available in LPC2144/46/48 only.
P0.13/DTR1/ MAT1.1/AD1.4	39 [4]	I/O	**P0.13** — General purpose input/output digital pin (GPIO).
		O	**DTR1** — Data Terminal Ready output for UART1. LPC2144/46/48 only.
		O	**MAT1.1** — Match output for Timer 1, channel 1.
		I	**AD1.4** — ADC input 4. Available in LPC2144/46/48 only.
P0.14/DCD1/ EINT1/SDA1	41 [3]	I/O	**P0.14** — General purpose input/output digital pin (GPIO).
		I	**DCD1** — Data Carrier Detect input for UART1. LPC2144/46/48 only.
		I	**EINT1** — External interrupt 1 input.
		I/O	**SDA1** — I^2C1 data input/output. Open-drain output (for I^2C-bus compliance) **Note:** LOW on this pin while $\overline{\text{RESET}}$ is LOW forces on-chip boot loader to take over control of the part after reset.
P0.15/RI1/ EINT2/AD1.5	45 [4]	I/O	**P0.15** — General purpose input/output digital pin (GPIO).
		I	**RI1** — Ring Indicator input for UART1. Available in LPC2144/46/48 only.
		I	**EINT2** — External interrupt 2 input.
		I	**AD1.5** — ADC 1, input 5. Available in LPC2144/46/48 only.
P0.16/EINT0/ MAT0.2/CAP0.2	46 [2]	I/O	**P0.16** — General purpose input/output digital pin (GPIO).
		I	**EINT0** — External interrupt 0 input.
		O	**MAT0.2** — Match output for Timer 0, channel 2.
		I	**CAP0.2** — Capture input for Timer 0, channel 2.
P0.17/CAP1.2/ SCK1/MAT1.2	47 [1]	I/O	**P0.17** — General purpose input/output digital pin (GPIO).
		I	**CAP1.2** — Capture input for Timer 1, channel 2.
		I/O	**SCK1** — Serial Clock for SSP. Clock output from master or input to slave.
		O	**MAT1.2** — Match output for Timer 1, channel 2.
P0.18/CAP1.3/ MISO1/MAT1.3	53 [1]	I/O	**P0.18** — General purpose input/output digital pin (GPIO).
		I	**CAP1.3** — Capture input for Timer 1, channel 3.
		I/O	**MISO1** — Master In Slave Out for SSP. Data input to SPI master or data output from SSP slave.
		O	**MAT1.3** — Match output for Timer 1, channel 3.
P0.19/MAT1.2/ MOSI1/CAP1.2	54 [1]	I/O	**P0.19** — General purpose input/output digital pin (GPIO).
		O	**MAT1.2** — Match output for Timer 1, channel 2.
		I/O	**MOSI1** — Master Out Slave In for SSP. Data output from SSP master or input to SSP slave.
		I	**CAP1.2** — Capture input for Timer 1, channel 2.
P0.20/MAT1.3/ SSEL1/EINT3	55 [2]	I/O	**P0.20** — General purpose input/output digital pin (GPIO).
		O	**MAT1.3** — Match output for Timer 1, channel 3.
		I	**SSEL1** — Slave Select for SSP. Selects the SSP interface as a slave.
		I	**EINT3** — External interrupt 3 input.
P0.21/PWM5/ AD1.6/CAP1.3	1 [4]	I/O	**P0.21** — General purpose input/output digital pin (GPIO).
		O	**PWM5** — Pulse Width Modulator output 5.
		I	**AD1.6** — ADC 1, input 6. Available in LPC2144/46/48 only.
		I	**CAP1.3** — Capture input for Timer 1, channel 3.
P0.22/AD1.7/ CAP0.0/MAT0.0	2 [4]	I/O	**P0.22** — General purpose input/output digital pin (GPIO).
		I	**AD1.7** — ADC 1, input 7. Available in LPC2144/46/48 only.
		I	**CAP0.0** — Capture input for Timer 0, channel 0.
		O	**MAT0.0** — Match output for Timer 0, channel 0.
P0.23/V_{BUS}	58 [1]	I/O	**P0.23** — General purpose input/output digital pin (GPIO).
		I	V_{BUS} — Indicates the presence of USB bus power. **Note:** This signal must be HIGH for USB reset to occur.
P0.25/AD0.4/ AOUT	9 [5]	I/O	**P0.25** — General purpose input/output digital pin (GPIO).
		I	**AD0.4** — ADC 0, input 4.
		O	**AOUT** — DAC output. Available in LPC2142/44/46/48 only.
P0.28/AD0.1/ CAP0.2/MAT0.2	13 [4]	I/O	**P0.28** — General purpose input/output digital pin (GPIO).
		I	**AD0.1** — ADC 0, input 1.
		I	**CAP0.2** — Capture input for Timer 0, channel 2.
		O	**MAT0.2** — Match output for Timer 0, channel 2.
P0.29/AD0.2/ CAP0.3/MAT0.3	14 [4]	I/O	**P0.29** — General purpose input/output digital pin (GPIO).
		I	**AD0.2** — ADC 0, input 2.
		I	**CAP0.3** — Capture input for Timer 0, Channel 3.
		O	**MAT0.3** — Match output for Timer 0, channel 3.
P0.30/AD0.3/ EINT3/CAP0.0	15 [4]	I/O	**P0.30** — General purpose input/output digital pin (GPIO).
		I	**AD0.3** — ADC 0, input 3.
		I	**EINT3** — External interrupt 3 input.
		I	**CAP0.0** — Capture input for Timer 0, channel 0.

Symbol	Pin	Type	Description
P0.31/UP_LED/ CONNECT	17 [6]	O	**P0.31** — General purpose output only digital pin (GPO).
		O	**UP_LED** — USB Good Link LED indicator. It is LOW when device is configured (non-control endpoints enabled). It is HIGH when the device is not configured or during global suspend.
		O	**CONNECT** — Signal used to switch an external 1.5 kΩ resistor under the software control. Used with the SoftConnect USB feature. **Important:** This is an digital output only pin. This pin MUST NOT be externally pulled LOW when RESET pin is LOW or the JTAG port will be disabled.
P1.0 to P1.31		I/O	**Port 1:** Port 1 is a 32-bit bidirectional I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 0 through 15 of port 1 are not available.
P1.16/ TRACEPKT0	16 [6]	I/O	**P1.16** — General purpose input/output digital pin (GPIO).
		O	**TRACEPKT0** — Trace Packet, bit 0. Standard I/O port with internal pull-up.
P1.17/ TRACEPKT1	12 [6]	I/O	**P1.17** — General purpose input/output digital pin (GPIO).
		O	**TRACEPKT1** — Trace Packet, bit 1. Standard I/O port with internal pull-up.
P1.18/ TRACEPKT2	8 [6]	I/O	**P1.18** — General purpose input/output digital pin (GPIO).
		O	**TRACEPKT2** — Trace Packet, bit 2. Standard I/O port with internal pull-up.
P1.19/ TRACEPKT3	4 [6]	I/O	**P1.19** — General purpose input/output digital pin (GPIO).
		O	**TRACEPKT3** — Trace Packet, bit 3. Standard I/O port with internal pull-up.
P1.20/ TRACESYNC	48 [6]	I/O	**P1.20** — General purpose input/output digital pin (GPIO).
		O	**TRACESYNC** — Trace Synchronization. Standard I/O port with internal pull-up. **Note:** LOW on this pin while RESET is LOW enables pins P1.25:16 to operate as Trace port after reset.
P1.21/ PIPESTAT0	44 [6]	I/O	**P1.21** — General purpose input/output digital pin (GPIO).
		O	**PIPESTAT0** — Pipeline Status, bit 0. Standard I/O port with internal pull-up.
P1.22/ PIPESTAT1	40 [6]	I/O	**P1.22** — General purpose input/output digital pin (GPIO).
		O	**PIPESTAT1** — Pipeline Status, bit 1. Standard I/O port with internal pull-up.
P1.23/ PIPESTAT2	36 [6]	I/O	**P1.23** — General purpose input/output digital pin (GPIO).
		O	**PIPESTAT2** — Pipeline Status, bit 2. Standard I/O port with internal pull-up.
P1.24/ TRACECLK	32 [6]	I/O	**P1.24** — General purpose input/output digital pin (GPIO).
		O	**TRACECLK** — Trace Clock. Standard I/O port with internal pull-up.
P1.25/EXTIN0	28 [6]	I/O	**P1.25** — General purpose input/output digital pin (GPIO).
		I	**EXTIN0** — External Trigger Input. Standard I/O with internal pull-up.
P1.26/RTCK	24 [6]	I/O	**P1.26** — General purpose input/output digital pin (GPIO).
		I/O	**RTCK** — Returned Test Clock output. Extra signal added to the JTAG port. Assists debugger synchronization when processor frequency varies. Bidirectional pin with internal pull-up. **Note:** LOW on RTCK while RESET is LOW enables pins P1.31:26 to operate as Debug port after reset.
P1.27/TDO	64 [6]	I/O	**P1.27** — General purpose input/output digital pin (GPIO).
		O	**TDO** — Test Data out for JTAG interface.

P1.28/TDI	60[6]	I/O	**P1.28** — General purpose input/output digital pin (GPIO).
		I	**TDI** — Test Data in for JTAG interface.
P1.29/TCK	56[6]	I/O	**P1.29** — General purpose input/output digital pin (GPIO).
		I	**TCK** — Test Clock for JTAG interface.
P1.30/TMS	52[6]	I/O	**P1.30** — General purpose input/output digital pin (GPIO).
		I	**TMS** — Test Mode Select for JTAG interface.
P1.31/$\overline{\text{TRST}}$	20[6]	I/O	**P1.31** — General purpose input/output digital pin (GPIO).
		I	**$\overline{\text{TRST}}$** — Test Reset for JTAG interface.
D+	10[7]	I/O	USB bidirectional D+ line.
D−	11[7]	I/O	USB bidirectional D− line.
$\overline{\text{RESET}}$	57[8]	I	**External reset input:** A LOW on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.
XTAL1	62[9]	I	Input to the oscillator circuit and internal clock generator circuits.
XTAL2	61[9]	O	Output from the oscillator amplifier.
RTXC1	3[9]	I	Input to the RTC oscillator circuit.
RTXC2	5[9]	O	Output from the RTC oscillator circuit.
V_{SS}	6, 18, 25, 42, 50	I	**Ground:** 0 V reference.
V_{SSA}	59	I	**Analog ground:** 0 V reference. This should nominally be the same voltage as V_{SS}, but should be isolated to minimize noise and error.
V_{DD}	23, 43, 51	I	**3.3 V power supply:** This is the power supply voltage for the core and I/O ports.
V_{DDA}	7	I	**Analog 3.3 V power supply:** This should be nominally the same voltage as V_{DD} but should be isolated to minimize noise and error. This voltage is only used to power the on-chip ADC(s) and DAC.
VREF	63	I	**ADC reference:** This should be nominally less than or equal to the V_{DD} voltage but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC(s) and DAC.
V_{BAT}	49	I	**RTC power supply:** 3.3 V on this pin supplies the power to the RTC.

[1] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control.

[2] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control. If configured for an input function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns.

[3] Open-drain 5 V tolerant digital I/O I^2C-bus 400 kHz specification compatible pad. It requires external pull-up to provide an output functionality.

[4] 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog input function. If configured for an input function, this pad utilizes built-in glitch filter that blocks pulses shorter than 3 ns. When configured as an ADC input, digital section of the pad is disabled.

[5] 5 V tolerant pad providing digital I/O (with TTL levels and hysteresis and 10 ns slew rate control) and analog output function. When configured as the DAC output, digital section of the pad is disabled.

[6] 5 V tolerant pad with built-in pull-up resistor providing digital I/O functions with TTL levels and hysteresis and 10 ns slew rate control. The pull-up resistor's value typically ranges from 60 kΩ to 300 kΩ.

[7] Pad is designed in accordance with the Universal Serial Bus (USB) specification, revision 2.0 (Full-speed and Low-speed mode only).

[8] 5 V tolerant pad providing digital input (with TTL levels and hysteresis) function only.

[9] Pad provides special analog functionality.

5. REAL MONITOR

Real Monitor is a configurable software module, developed by ARM Inc., which enables real-time debug. It is a lightweight debug monitor that runs in the background while users debug their foreground application. It communicates with the host using the DCC, which is present in the Embedded ICE logic. The LPC2141/42/44/46/48 contains a specific configuration of Real Monitor software programmed into the on-chip flash memory.

ARM7 LPC2148 is ARM7TDMI-S Core Board Microcontroller that uses 16/32-Bit 64 Pin (LQFP) Microcontroller No.LPC2148 from Philips (NXP). All resources inside LPC2148 is quite perfect, so it is the most suitable to learn and study because if user can learn and understand the applications of all resources inside MCU well, it makes user can modify, apply and develop many excellent applications in the future. Because Hardware system of LPC2148 includes the necessary devices within only one MCU such as USB, ADC, DAC, Timer/Counter, PWM, Capture, I2C, SPI, UART, and etc.

Board Technical Specifications

Processor* : LPC2148

Clock speed : 11.0592 MHz / 22.1184 MHz

Clock Divisors : 6 (or) 12

Real time Clock : DS1307 on i2c Bus /w Battery

Data Memory : 24LCxx on i2c Bus

LCD : 16x2 Backlight

LED indicators : Power

RS-232 : +9V -9V levels

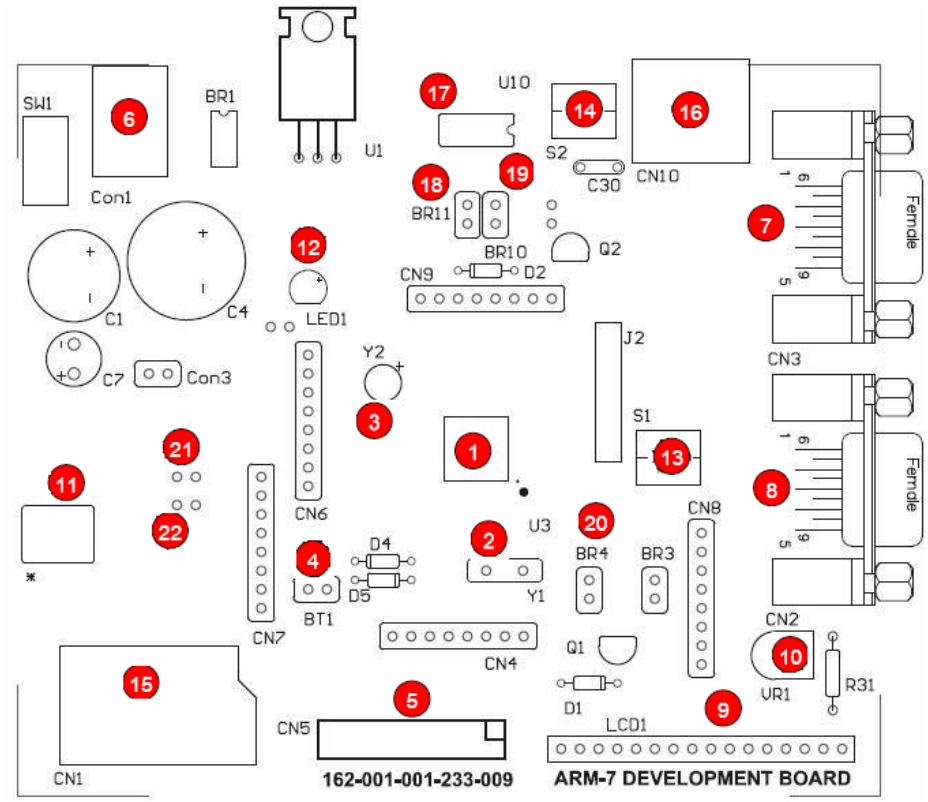
Power : 7-15V AC/DC @ 500mA

Voltage Regulator : 5V Onboard LM7805

Specifications of Board:

- Use 16/32 Bit ARM7TDMI-S MCU No.LPC2148 from Philips (NXP) has 512KB Flash Memory and 40KB Static RAM internal MCU use 12.00MHz Crystal, so MCU can process data with the maximum high speed at 60MHz when using it with Phase-Locked Loop (PLL) internal MCU has RTC Circuit (Real Time Clock) with 32.768 KHz XTAL and Battery Backup.
- Support In-System Programming (ISP) and In-Application Programming (IAP) through On-Chip Boot-Loader Software via Port UART-0 (RS232) has circuit to connect with standard 20 Pin JTAG ARM for Real Time Debugging 7-12V AC/DC Power Supply.
- Has standard 2.0 USB as Full Speed inside (USB Function has 32 End Point)
- Has Circuit to connect with Dot-Matrix LCD with circuit to adjust its contrast by using 16 PIN Connector.
- Has RS232 Communication Circuit by using 2 Channel.
- Has SD/MMC card connector circuit by using SSP.
- Has EEPROM interface using I2C.
- Has PS2 keyboard interface.
- All port pins are extracted externally for further interfaces.

ISSN: 2278-6252



BOM of LPC2148

BOM OF LPC2148 Board

- **No 1 is** MCU No.LPC2368 (100Pin LQFP).
- **No.2 is** 12MHz Crystal to be Time Base of MCU.
- **No.3** is 32.768 KHz Crystal to be Time Base of RTC internal MCU.
- **No.4** is 3V Battery for Backup of RTC.
- **No.5** is JTAG ARM Connector for Real Time Debugging.
- **No.6** is Power Supply Connector of board; it can be used with 7-12V AC/DC.
- **No.7** is UART-0(RS232) Connector to use and Download Hex File into CPU.
- **No.8** is UART-2(RS232) Connector to use.
- **No.9** is Character LCD Connector; it can be used with +5V Supply LCD.
- **No.10** is VR to adjust the contrast or brightness of Character LCD.
- **No.11** is USB Connector to connect with USB Hub version 2.0.
- **No.12** is LED to display status of Power +VDD (+3V3).
- **No.13** is S1 that is ISP LOAD.

- **No.14**is S2 or RESET Switch.
- **No.15** is socket to insert Memory Card; it can be used with both SD Memory Card and MMC Memory Card.
- **No.16** is PS2 Connector to connect with PS2 keyboard.
- **No.17** is External Memory.
- **No.18** and **No.19** is jumper to connect External Memory to MCU.
- **No.20** is jumper to connect INT1.
- **No.21** and **No.22** is jumper to connect D- & D+ to the USB connector.

Jumper Settings for Interfaces:

Jumper	*State*	*Description*
BR10 – SCL	ON	Connects I2C SCL to EEPROM
BR11 – SDA	ON	Connects I2C SDA to EEPROM
BR5 – USB (D-)	ON	Connects USB Line D- to the USB connector
BR6 – USB (D+)	ON	Connects USB Line D+ to the USB connector
BR2 – Vbus	ON	Connects 5V USB supply voltage to the Vbus pin

Implementation

Software Required

The software required for this project to develop is as following:

Windows XP

This is an Operating System (OS) on which all the software applications required for our project are going to be run. This OS is flexible to any user to operate and easy to understood. Accessing the soft wares and using them is very convenient to user.

ORCAD

ORCAD is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly to create electronic prints for manufacturing of printed circuit boards, by electronic design engineers and electronic technicians to manufacture electronic schematics.

Keil Micro vision 3 IDE

The µVision development platform is easy-to-use and it helps you quickly create embedded programs that work. The µVision IDE (Integrated Development Environment) from Keil

combines project management, source code editing, program debugging, and complete simulation in one powerful environment. Code written in 'EMBEDDED C'

Software Development Cycle

When you use the Keil µVision3, the project development cycle is roughly the same as it is for any other software development project.

1. Create a project, select the target chip from the device database, and configure the tool settings.
2. Create source files in C or assembly.
3. Build your application with the project manager.
4. Correct errors in source files.
5. Test the linked application.

The following block diagram illustrates the complete µVision3/ARM software development cycle. Each component is described below

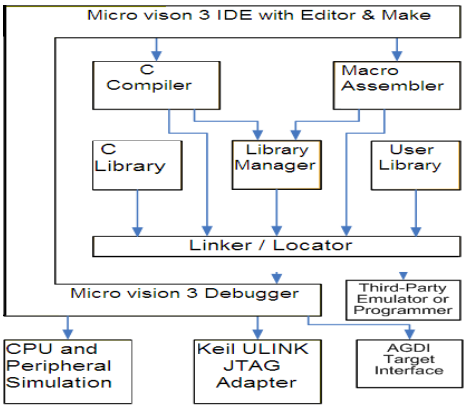


Figure 12: Software Development cycle in keil Software

In the µVision3 Debug Mode you verify your program either with a powerful CPU and peripheral simulator or with the Keil ULINK USB-JTAG Adapter (or other AGDI drivers) that connect the debugger to the target system. The ULINK allows you also to download your application into Flash ROM of your target system.

Flash Magic Software

How to Download Hex File into MCU of Board

The method to download Hex File into Flash Memory of MCU in Board is to use Program Flash Magic that is connected with MCU through Serial Port of computer PC. This program can be downloaded free without any charge from website http://www.flashmagictool.com/

Proceeding to Download Hex File into MCU

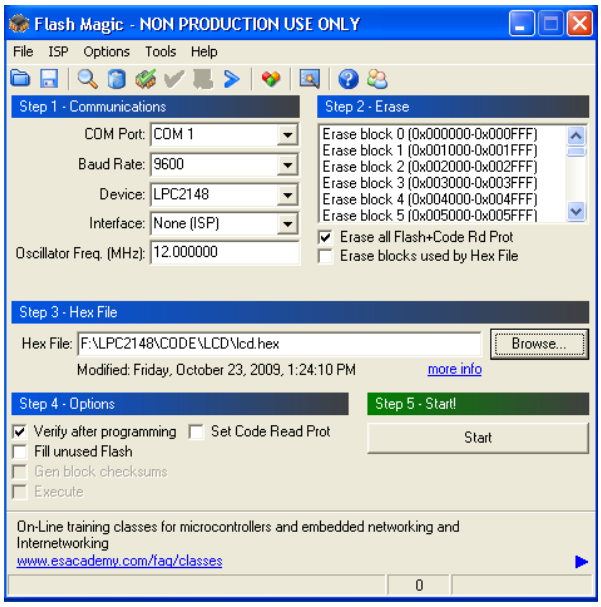


Figure 13 :Flash Magic Software

6. CONCLUSIONS

This Paper presents a Implementation of "Industrial Object Sorting Robot". The Paper is been designed and implemented with embedded system domain using IR communication. Experimental work has been carried out carefully. The proposed method is verified to be highly beneficial for automated industries. This is an excellent application for a robot of this type. In this case, to keep costs and design complexity low, the robot is designed around the platform and uses several different sensors to collect information about the robots environment to allow the robot to react accordingly.

REFERENCES

1. David Seal ARM Architecture Reference Manual: Addison-Wesley
2. Steve Furber ARM System-on-chip Architecture
3. Dr.R.K Bansal , A Text Book of theory of Machines
4. Mikell P. Groover, Industrial Automation and Robotics
5. R.S Kurmi, Theory of Machines
6. Michael J.pont, Embedded c
7. www.howstuffworks.com
8. www.answers.com
9. www.radiotronix.com
10. www.WineYardProjects.com
11. www.Wikipedia.com