Introduction to Engineering Design

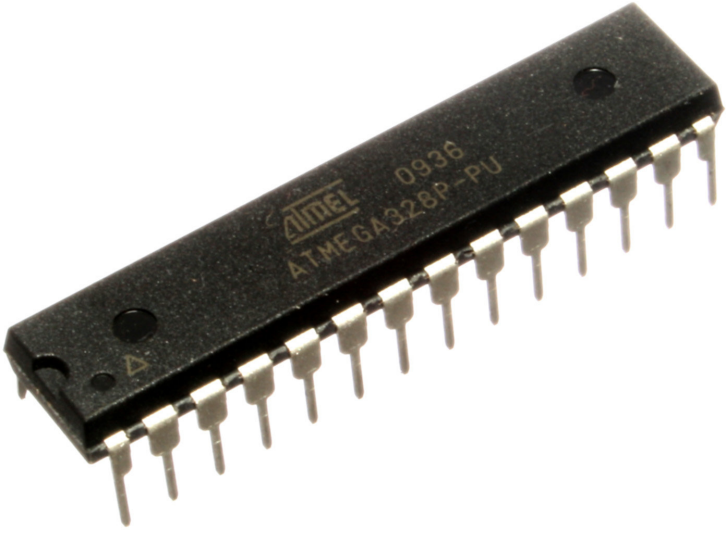
Microcontroller Units, Communication Protocols

Alexander Fell
alex@iiitd.ac.in

January 16th, 2015

Outline

What is a microcontroller?

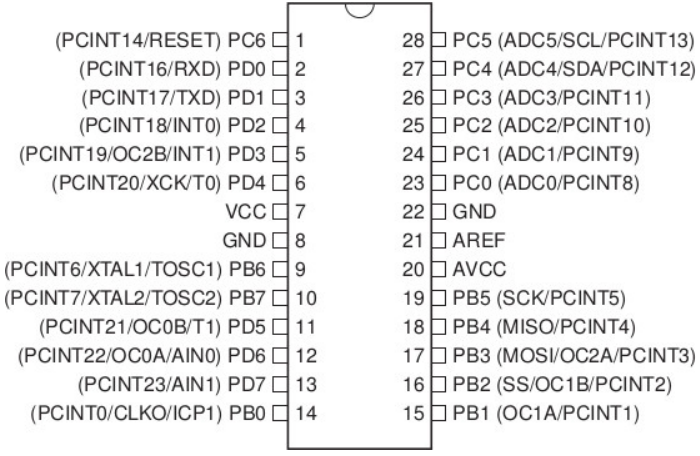


ATmega328P-PU

Specifications I

- $\leq$ 20MHz
- 2KB RAM
- 32KB Flash
- Supply Voltage: 1.8V - 5.5V
- Power: 45mW (5V @ 8MHz), with power saving techniques 0.015mW (3V)

Specifications II



Source: [3]

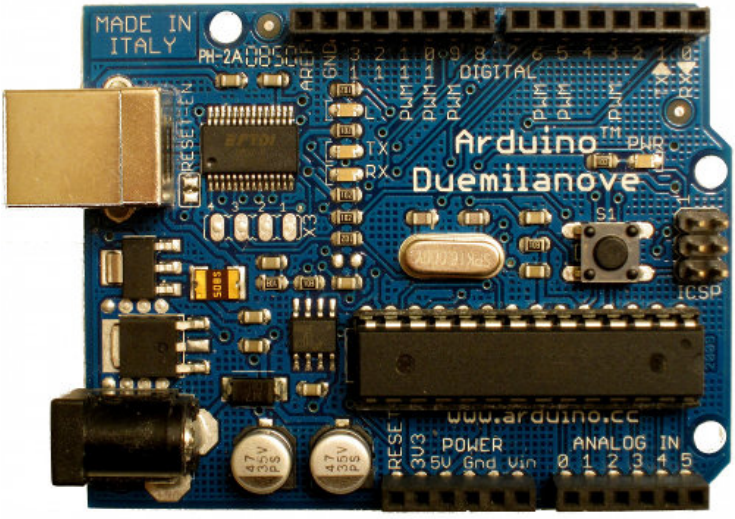
- 3 ports: PORT(B|C|D)

Hardware support for

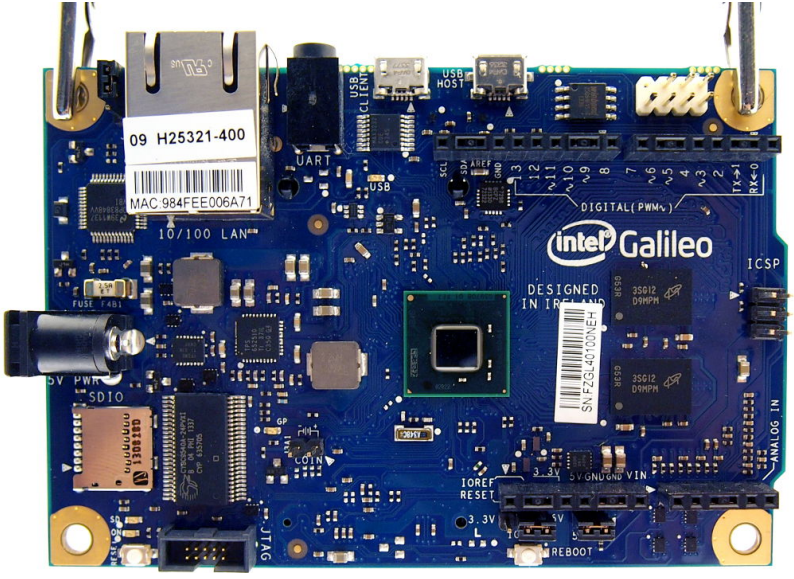
- UART/USART, I^2C (TWI), SPI
- PWM
- ADC

What is a Microcontroller Unit?

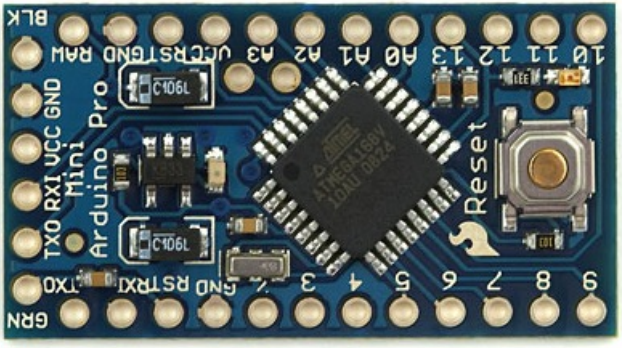
What is a Microcontroller Unit?



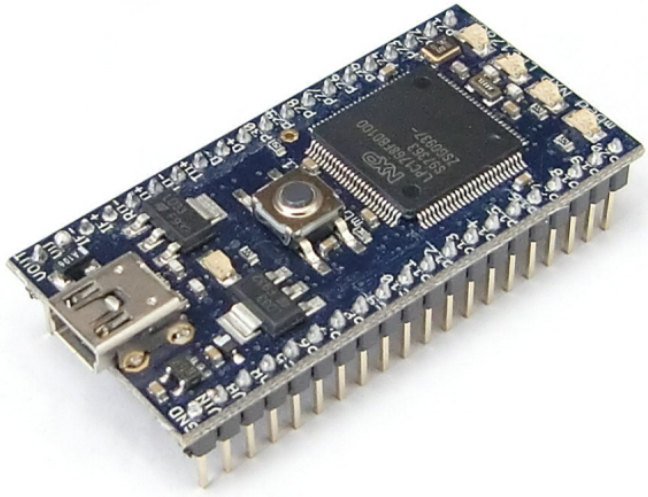
A few other MCUs I



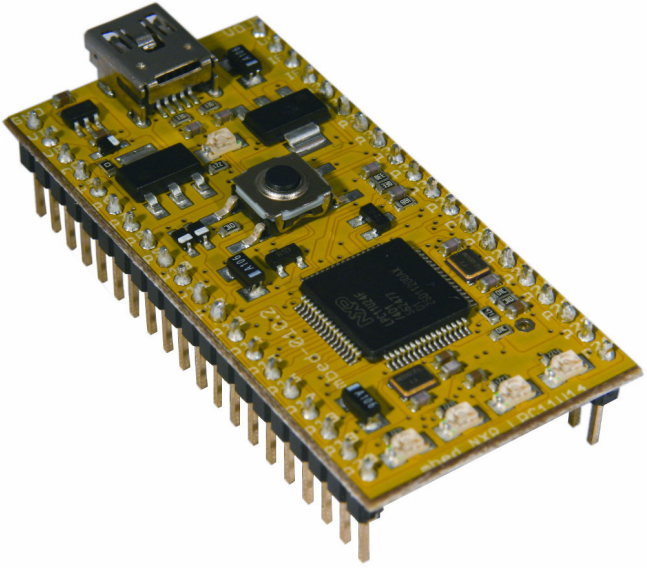
A few other MCUs II



A few other MCUs III

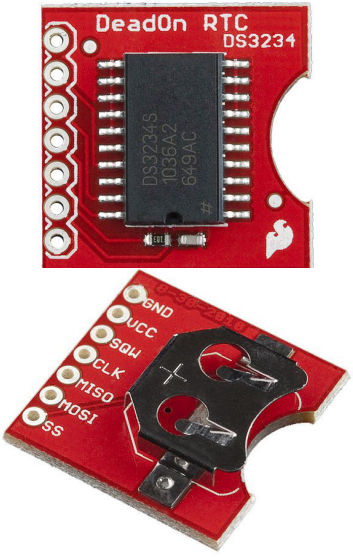


A few other MCUs IV

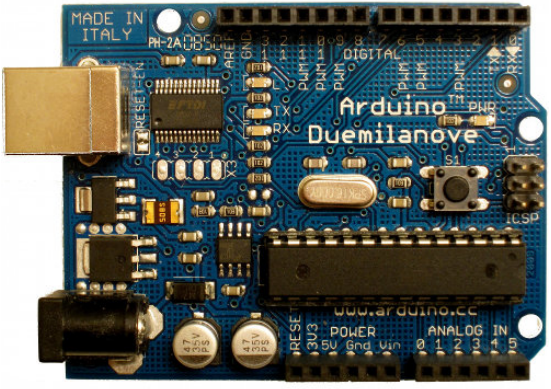


Where to use MCUs?

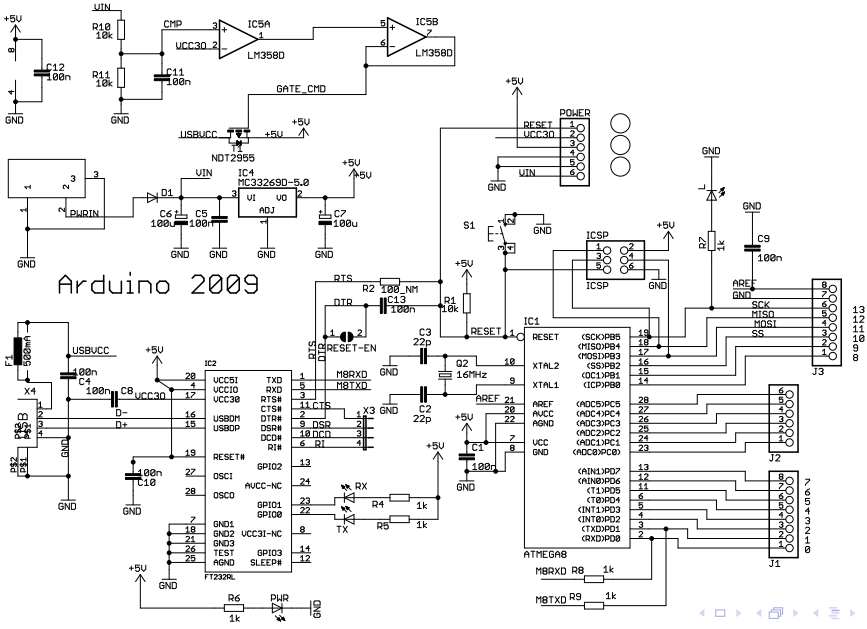
How to connect the RTC to the Arduino?



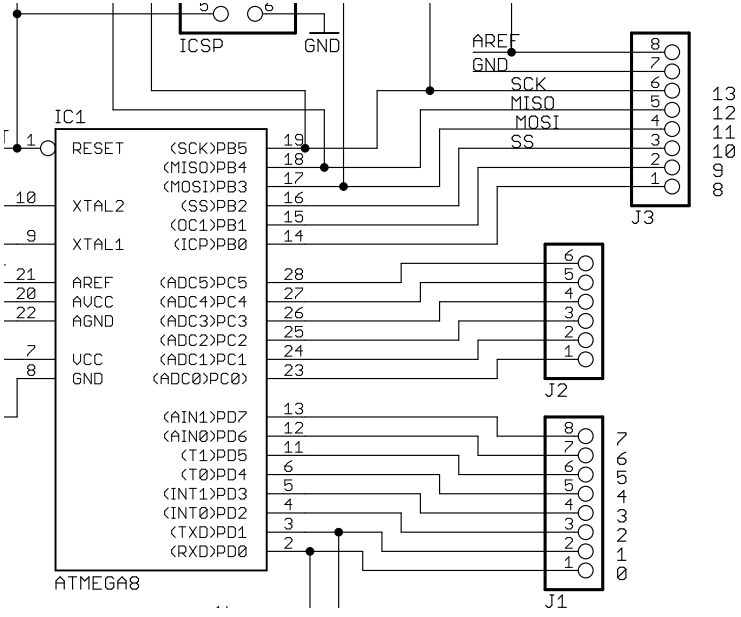
RTC DS3231 front and back



Arduino Duemilanove Schematics I



Arduino Duemilanove Schematics II



Protocols

There are many communication protocols out there. How to choose one? Which aspects do you need to consider?

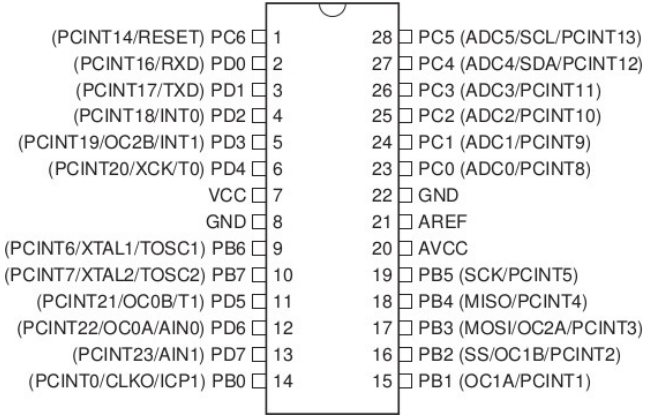
Outline

Specifications

- Common way to interconnect devices within embedded systems
- In existence for more than 30 years, invented by Philips in 1982
- Data rate: 100 kbps (v1, standard mode), 400 kbps (v2, fast mode)
  - Most current: v4 (2012), with 5Mbps peak data rate
- Two wire interface: SDA (Serial Data) and SCL (Serial Clock)
- Bus is half-duplex.
- Length limited by capacitance (400pF), a few meters only.
- Every device has a unique address within the bus

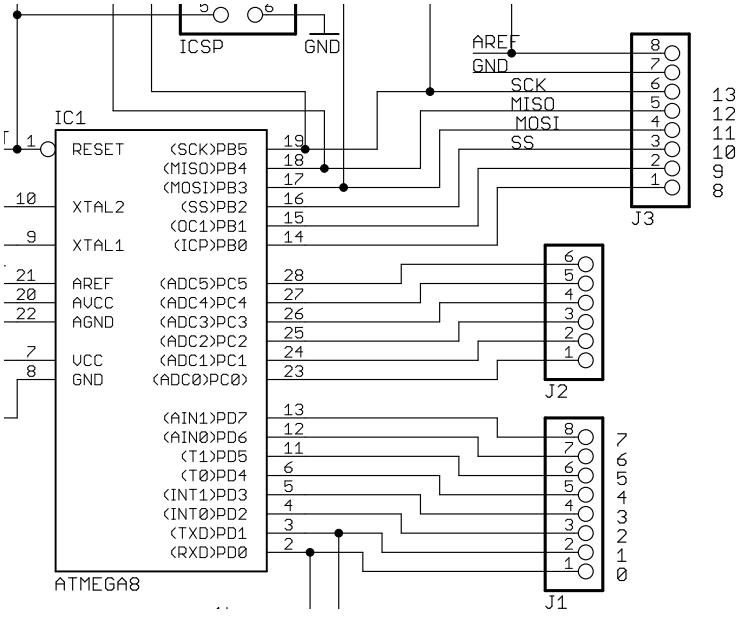
Where is TWI/I^{2}C? I

- 1 wire for a clock signal: SCL (Serial Clock)
- 1 wire for the data signal: SDA (Serial Data)

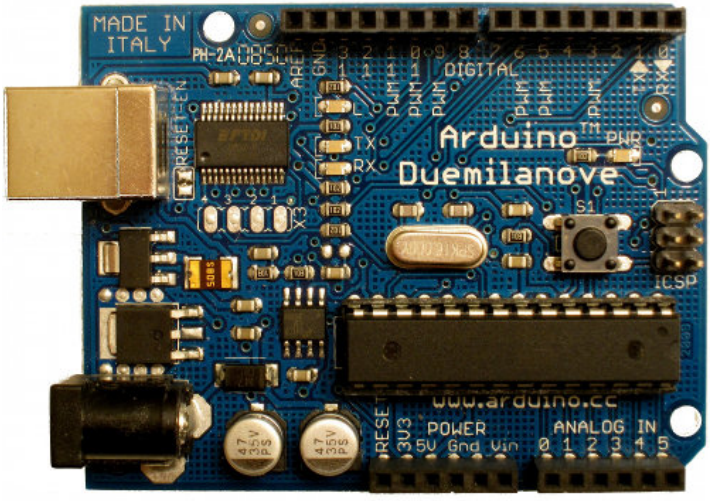


Source: [3]

Where is TWI/I^{2}C? II



Where is TWI/I^{2}C? III



The Bus I

Validity of Data

TWI/I^{2}C Packet

TWI/I^{2}C Key Observations

Serial Peripheral Interface (SPI)

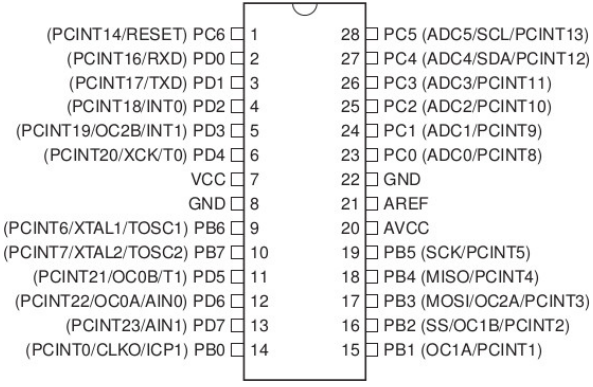
- Invented by Motorola in 1979
- No formal specification available

Serial Peripheral Interface (SPI)

- Invented by Motorola in 1979
- No formal specification available $\rightarrow$ data sheets reveal supported modes for devices[4]
  - Might not be overlapping between devices
- Wires ≥ 2, usually 4 (SCK, MISO, MOSI, SS)
- Synchronous protocol: All transmissions referenced to common clock generated by the master (e.g. processor)
  - Receiver (slave) uses the clock to synchronize the bit stream
- Slave addressed by chip select line
- Single master device and multiple slave devices
- Used for a variety of peripherals:
  - Sensors: temperature, pressure
  - Control devices: audio codecs, digital potentiometer
  - Memory: Flash, EEPROM, SD card
  - Real time clocks
  - LCD displays

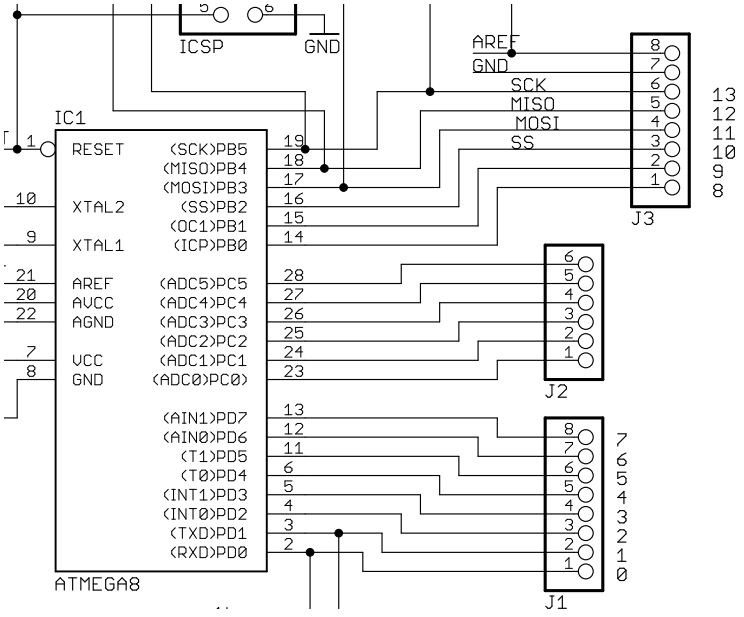
Where is SPI? I

- 1 wire for a clock signal: SCK
- 2 wires for the data signal: MISO, MOSI
- 1 wire for chip/slave select: SS

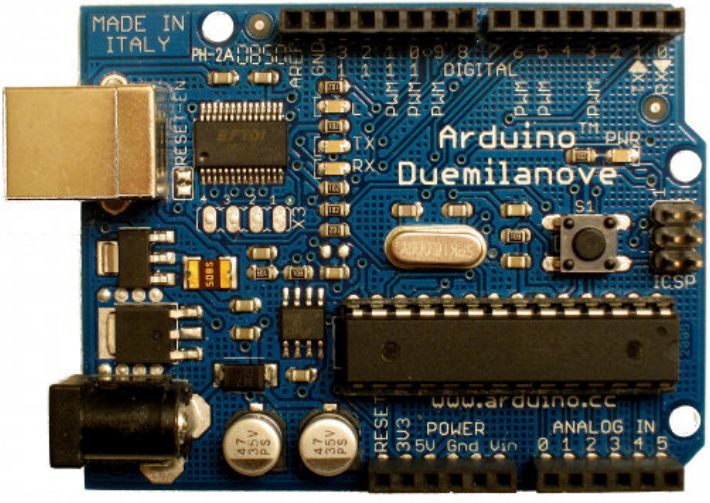


Source: [3]

Where is SPI? II



Where is SPI? III



SPI Signals - 4 Wires I

Internal Blocks[1, 5]

Option 1

Option 2

SPI Signals - 2/3 Wires

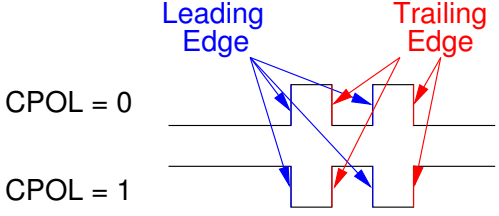
SPI Signals - Many Wires

Four modes of operation depending on clock polarity and clock phase

- Clock polarity:

$$\text{CPOL} = \begin{cases} 0, & \text{SCK is low, when idle} \\ 1, & \text{otherwise} \end{cases}$$

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

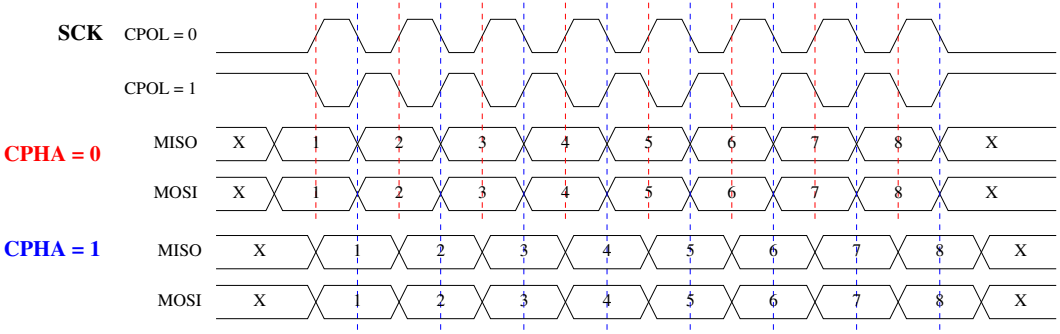


- Clock phase:

$$\text{CPHA} = \begin{cases} 0, & \text{MOSI/MISO are valid on leading edge of SCK} \\ 1, & \text{otherwise} \end{cases}$$

CPHA	**Leading Edge**	**Trailing Edge**
0	Sample	Setup
1	Setup	Sample

Communication Protocol of SPI III



SPI: timing diagram depicting 4 operation modes[2]

Modes	CPOL	CPHA
0*	0	0
1	0	1
2	1	0
3*	1	1

*Most commonly used modes

SPI Key Observations

- Invented by Dallas Semiconductor Corp. (now a subsidiary Maxim Integrated)
- Uses 1 wire for communication and powering slave devices
- Each device has a unique 64bit ID built in
- Data rate: 15.4kbps (standard), 125kbps (overdrive)
- Distances: $\leq$ 300m
- Half duplex
- No hardware support on Arduino (available through software)

- Most common application: iButton for access control, in IED: DHT11

iButton, photographer: Stan Zurek

The Bus

Data Transfer

Further Information

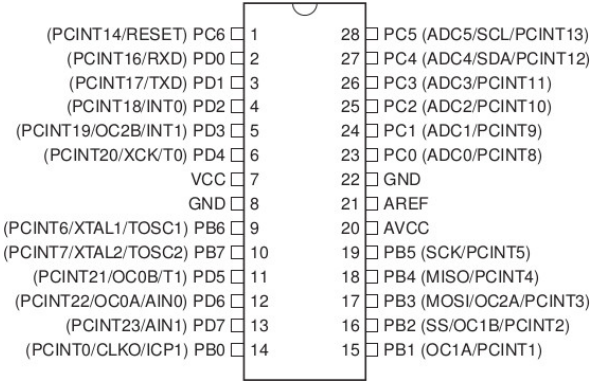
- Tutorial by Maxim Integrated
  - `http://www.maximintegrated.com/products/1-wire/flash/overview/index.cfm`
- Also available for download in `http://usebackpack.com`
  → Resources
  - Open `start.html` in a browser

Universal **A**synchronous **R**eceiver/**T**ransmitter

- I/O interface converting parallel data from microprocessor for serial communication and vice versa
- *Universal*: data format/transmission speeds are configurable (120Bd to 4mBd, here bd = bit/s)
- Independent transmitter/receiver
- Transmitter: 1 wire (TX), Receiver: 1 wire (RX) - No common bus system for both (refer to I^2C and 1-Wire)
- A simple data integrity check (parity) at receiver, if configured
- USART: It is a UART, that in addition can also communicate synchronously by adding a clock (XCK)

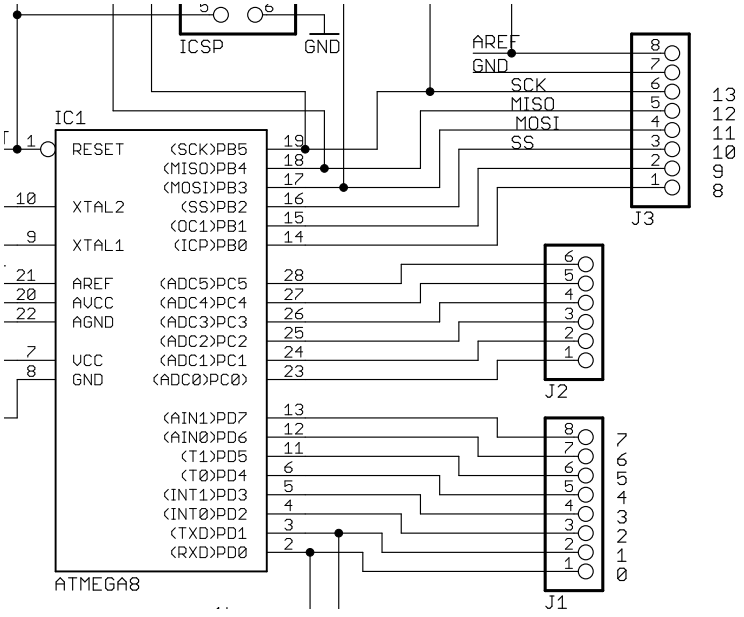
Where is UART/USART? I

- 1 wire for transmitter: TX
- 1 wire for receiver: RX
- (1 wire for clock: XCK)

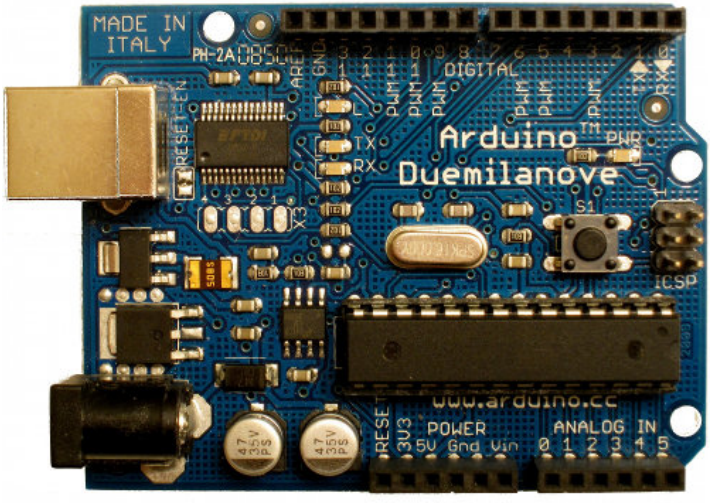


Source: [3]

Where is UART/USART? II



Where is UART/USART? III



UART Packet

Parity Bit I

Parity Bit II

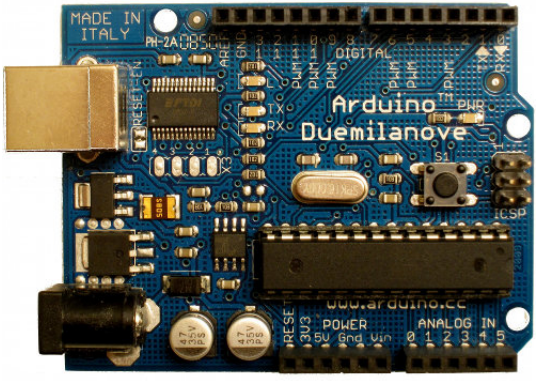
Outline

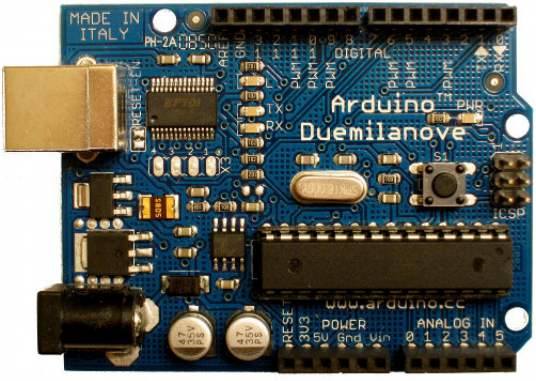
Comparison SPI, I^2C/TWI, 1-Wire, UART/USART

Create a table comparing the following properties of the discussed communication protocols:

- Wires required
- Duplex
- Broadcast/Multicast
- Hot plugging
- Number of supported master devices
- Number of slaves
- Clock speed (throughput, speed, bandwidth)
- Protocol overhead
- Error checking available?
- Acknowledgements
- Distances that can be covered

Connect both Arduinos using all protocols discussed before!





Outline

References I

[1] AVRBeginners.net.
The Serial Peripheral Interface (SPI).
http://avrbeginners.net/architecture/spi/spi.html.
Accessed: 2013-08-30.

[2] Santanu Chattopadhyay.
Embedded Systems Design.
PHI Learning, 2013.

[3] Atmel Inc.
ATmega48A/PA/88A/PA/168A/PA/328/P Complete.
http:
//www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf.
Accessed: 2013-08-06.

[4] Byte Paradigm.
Introduction to I^2C and SPI protocols.
http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/.
Accessed: 2013-08-30.

[5] Joshua R. Smith.
Serial Peripheral Interface.
University of Washington, 2011.
http://www.cs.washington.edu/education/courses/cse466/11au/calendar/07-comms-posted2.pdf; accessed 2013-08-30.