Week 9:
Connecting Systems

James Irvine

University of Strathclyde Engineering

DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

Interface Options

- Parallel
- Serial

- Point to Point
- Bus

Parallel vs Serial

- Parallel (bit parallel, word serial)
  - Higher speed for a given transmission medium
  - Handshaking still required for words
  - Conceptually simple, but high hardware cost
  - Used for short ranges where very high speed required, or
  - Resource rich devices where complexity is an issue
- Serial
  - Much lower hardware costs
  - More loading on software
  - Slower speed in theory, but bus speed usually not a constraint
  - Used everywhere

Network Structure

- Point to point
  - Simple
  - No contention
  - Significantly increased hardware/delays for >2 devices
- Bus
  - Very flexible
  - Speed independent of number of devices
  - Additional control hardware and software necessary for contention

In band control vs
Out of band control

- In band control

  Signalling messages to control the media (start/stop/ready/etc) is sent with the data on the same media

- Out of band control

  Signalling messages to control the media is sent with on different media (i.e. in parallel) with the data (i.e. handshaking lines)

- Out of band needs less software/more hardware

Synchronous vs Asynchronous

- Synchronous

  All information is sent with reference to an external clock, transmitted separately
- Asynchronous

  Symbol timing is inherent in the data stream and there is no external clock

- Synchronous requires more hardware, but is conceptually simpler and can be faster

Unidirectional vs Bidirectional

- Unidirectional

  Signals in the channel can only travel in one direction
- Bidirectional

  Signals in the channel can travel in both directions

- Bidirectional requires fewer lines but more handshaking

Parallel

- ☺ Really simple
- ☹ Uses up too many I/O lines
- Tends to have niche applications
  - Chips, printers, displays
  - Sometimes 8/4 bit option to save IO (as in the LCD display)
- Very expensive if long distances
- Very difficult if high speed
- Examples
  - Centronics (point to point)
  - IEEE 448 (bus)

Wired Serial Options

- Clocked serial
  - Very simple, but have to build both ends
- I^2C
  - Almost as simple; Driver chips and peripherals available
  - Allows connection of complex peripherals with few IO lines
- RS232
  - Low complexity; Allows connection to PCs, but usually through a USB to serial interface nowadays
- CAN
  - Bus structure with priorities
- USB
  - Very complex software, but ubiquitous from a PC viewpoint
- Ethernet
  - Even more complex software + hardware, but driver chips available

RS232

- As near a standard as there is
- Hardware support on most micros
- Often used to communicate with complex peripherals
  - GPRS Modems, etc
- Often now encapsulated over other media
  - USB <> serial chipsets
  - Bluetooth <> serial chipsets
- ☹ Point to point (unless you implement token passing or a MAC)
- RS485 is similar but less common – 32 devices, serial bus
  - Common on power industry devices, although being replaced by Ethernet

I^2C interface (Inter-IC, IIC)

- Developed originally by Phillips
- Many peripherals available (memory, sensors, etc)
- Only two lines – clock and data
- Master/slave arrangement
- Open collector connection, any device can pull line low
- Idle state is when line high (no device connecting line low)
- REMEMBER THE ADD PULL-UPS
  - Or the line will never got high!

I^{2}C Architecture

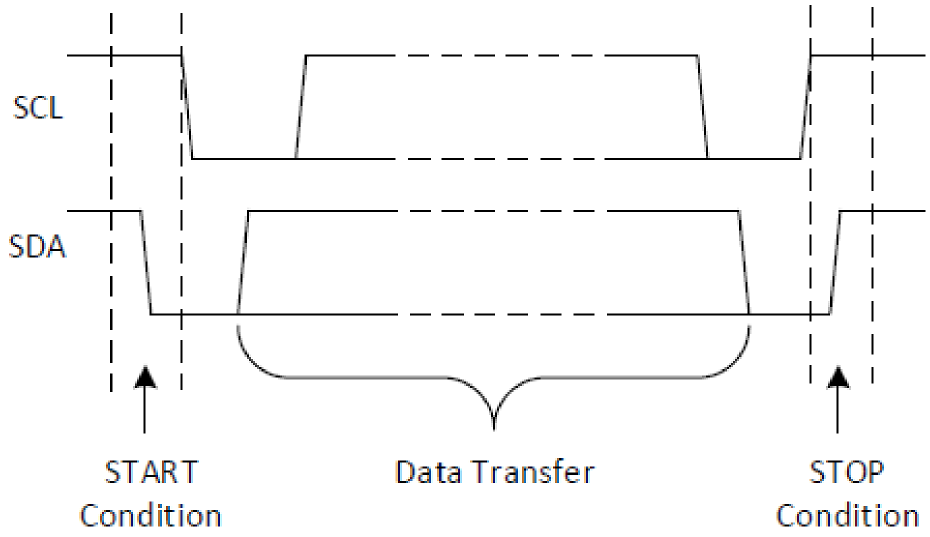
- Master device controls all data flow
- Data sent in 8 bit bytes
- Devices have 7 bit addresses
- Masters sends/receives to devices by sending their address and a data direction bit
  - R/W – 1 for read from peripheral to master, 0 send from master
- Receiver sends one bit acknowledgement
- After sending address, master sends register address
  - Not all peripherals have register addresses – not necessary if only one address on chip

I^2C Operation

- To send:
  - Assert START condition
  - Send data
  - Peripheral brings line low as an ACK (if successful)
  - Assert STOP
- To receive:
  - Master asserts START
  - Sends peripheral address
  - Peripheral asserts data
  - Master brings line low if appropriate for ACK
    - Master sends NACK to stop multi-byte transfer
  - Master asserts STOP

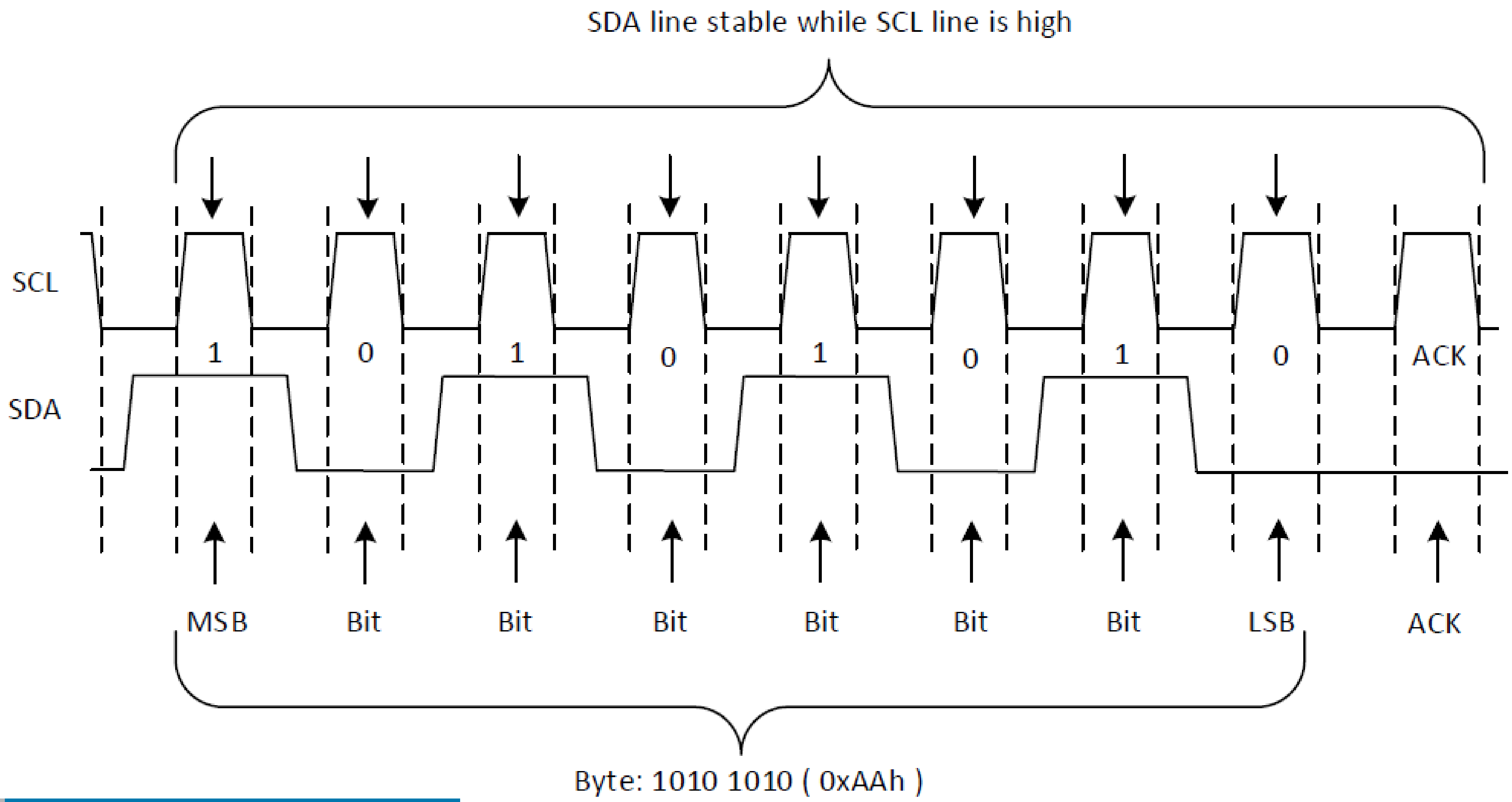
I^{2}C Signalling

- Sending data
  - Data line taken high or low kept there
  - Clock line taken high, then low
  - Note **data line unchanged when clock high**
- START
  - Data line taken low when clock line is high
- STOP
  - Data line taken high when clock high

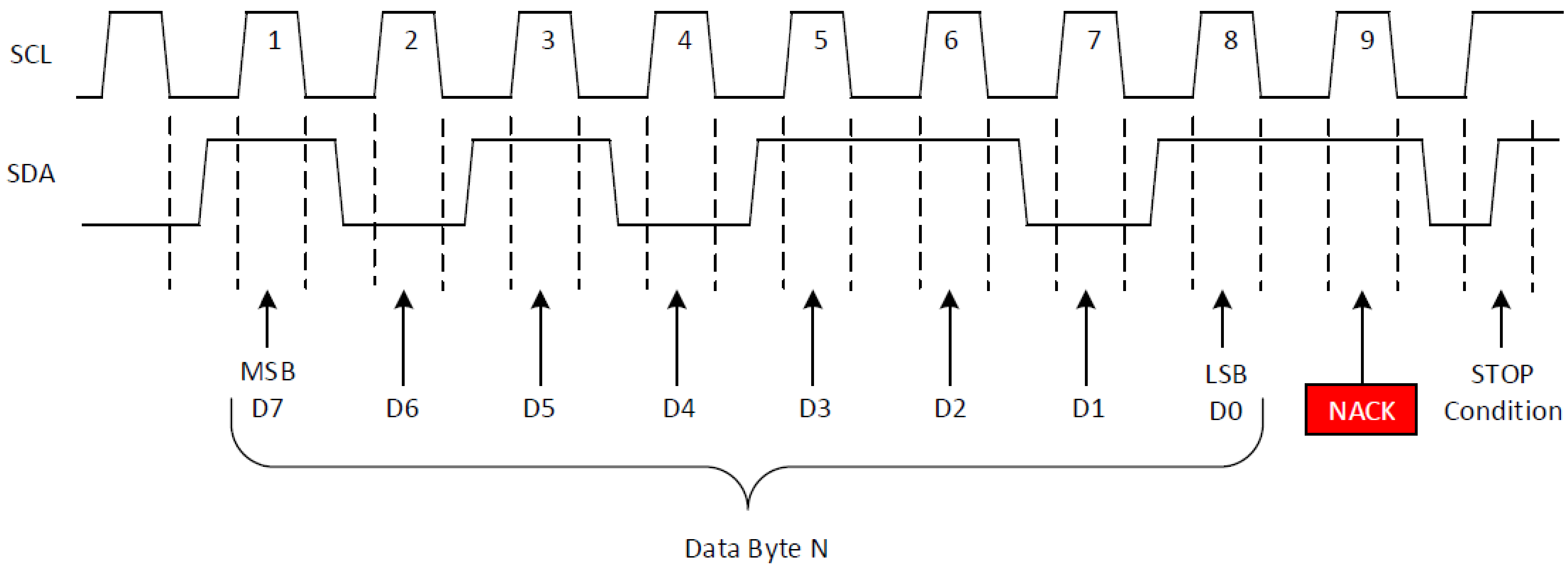


I^2C Signalling (successful ACK)

University of Strathclyde Engineering



I^{2}C Signalling (unsuccessful)



CAN (Controller Area Network)

- Two wire physical layer
- Started in automotive sector
  - Also used in industry control applications (e.g. DeviceNet)
- Bus structure, messages broadcast
- No master
  - Carrier Sense Medium Access with Collision Detection and Arbitration of Message Priority (CSMA / CD + AMP)
  - Small messages
- Wire ORed – lower IDs get priority
- Complex physical layer
  - Driver chips are available

USB

- Fast (up to 480Mbps!)
- Very low hardware requirement
- Ubiquitous PC/smartphone/etc connection
- Host/client structure
  - Most functionality in host, keeps clients ‘simple’
- Flexible data exchange
  - Control
  - Bulk data
  - Interrupt data
  - Isochronous (streaming)

USB

- ☺ Great for bulk data / PC peripherals
- ☺ Simple interfaces can be bit banged (just!)
- ☹ Much more complex than RS232

- Practical solutions
  - Get a USB driver chip, or microcontroller with built in USB controller (there are even PICs with USB – e.g., PIC18F2550 for £3.50 1 off)
  - If simply connecting to a PC for control, get a USB to serial chip

Ethernet

- ☺ Hugely flexible
  - Control your system from anywhere
- ☹ Hugely complex
  - IP stack required in practice (watch licensing)
  - Ethernet driver chips required

- Having said that, if a PIC can do it…

Wireless

- Advantages
  - No infrastructure
  - Simple deployment
- Disadvantages
  - More complex
  - More expensive
  - Multiple access, so may be contention
- Power has to be provided anyway

Wireless Options

University of Strathclyde Engineering

- Proprietary
  - ☺ Cheap (hardware and software)
  - ☺ Simpler
  - ☹ Not universal
- Wi-Fi
  - ☺ Universal
  - Used to be extremely complex
  - Getting cheaper (2014: CC3000 ~$10; 2020: ESP-01S $1)
  - Still an issue to set up (encryption, etc, although again solutions exist, eg, Simplelink)
    - User involvement, so not plug and play
  - ☹ High power requirements

Other Short Range Wireless Options

- Bluetooth
  - More expensive/complex than it should be
  - Okay with dedicated chips (often serial <> Bluetooth with no programming)
  - Universal availability with laptops/phones
  - BLE has very low power reqs, but watch compatibility
- Zigbee
  - What it's designed for, but slow take-up (Smart Meters though)
  - Being overtaken by BLE

Longer Range Wireless Options

University of Strathclyde Engineering

- LoRa
  - Cheap, unlicensed but commercial IoT networks available in some areas
  - Range of several kilometres for point to point
  - Modules $10+
  - Beware proprietary alternatives (SigFox) – companies go bust
- Cellular
  - ☺ Long range
  - ☹ Expensive ($50)
  - Traditionally GPRS but this is going away
  - 4G offers a range of solutions (NB-IoT, LTE-M, etc)
  - Use it when you can't rely on other infrastructure

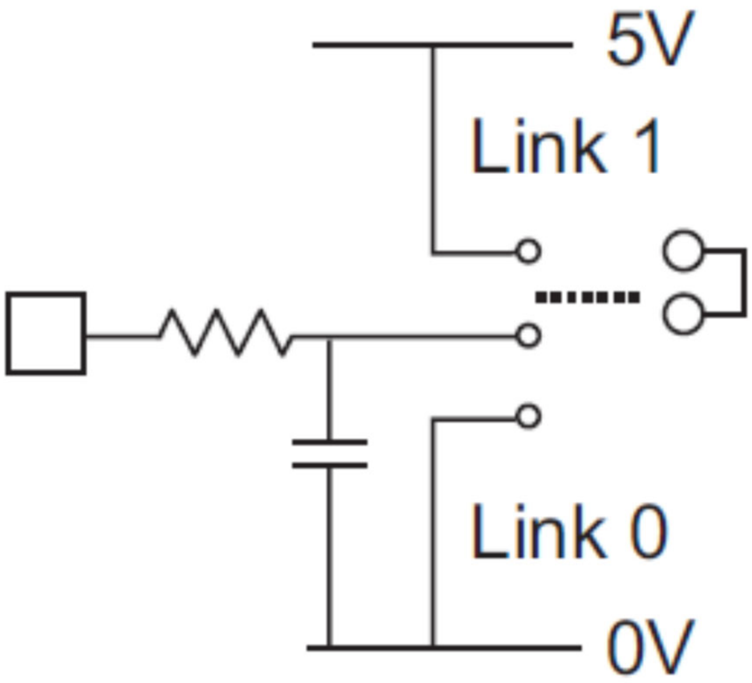
PIC Tips 'n Tricks

- PICs have very constrained IO (down to 4 pins on a 6 pin device)
- But – very cheap ☺
- Lots of good ideas for doing more with less

  http://ww1.microchip.com/downloads/en/devicedoc/40040b.pdf

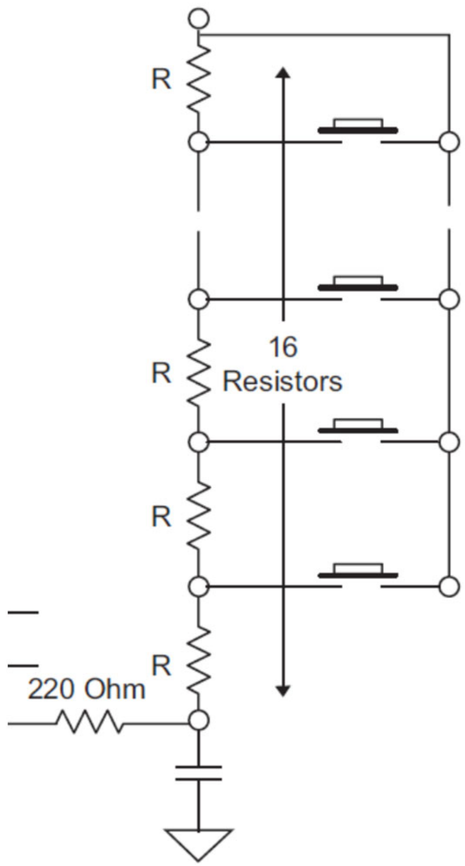
Read Three States From One Pin

- To check state Z:
  - Drive output pin high
  - Set to Input
  - Read 1
  - Drive output pin low
  - Set to Input
  - Read 0
- To check state 0:
  - Read 0 on pin
- To check state 1:
  - Read 1 on pin



Multiple Keys with Single Digital Pin

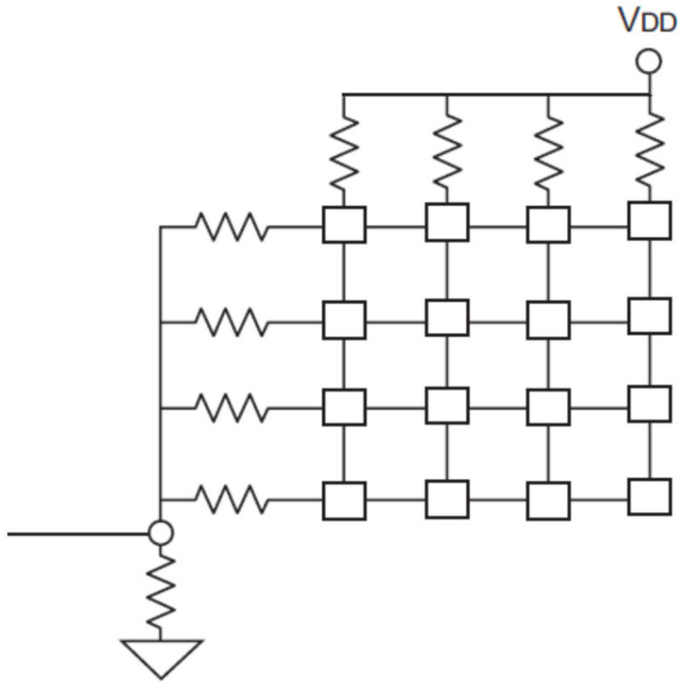
- Variable resistance can be read with an RC circuit
- Program pin as a low output
  - Capacitor discharges
- Program as an input
  - Initially pin will read high
  - Will go high after a time determined by the switch pressed
- Can use the same principle to read thermistors, etc with a digital pin



4x4 Keyboard with Analogue Pin

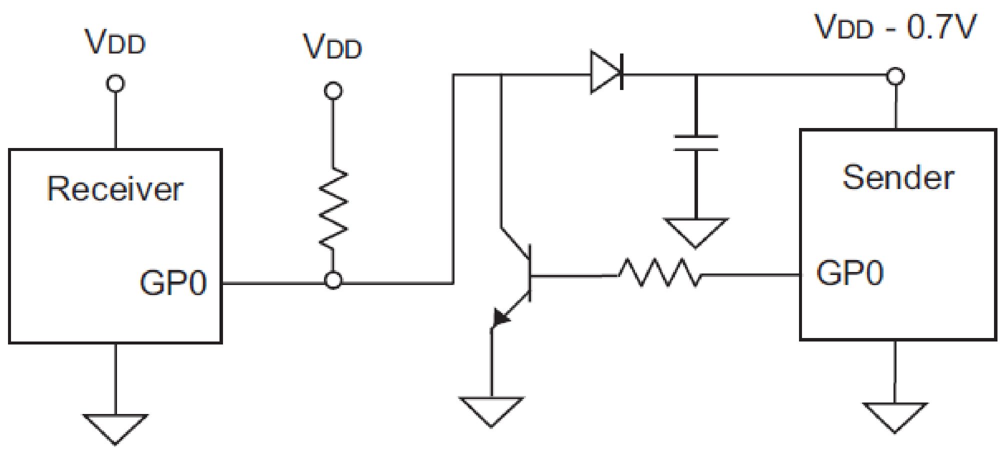
University of Strathclyde Engineering

- Pin = 0 if no button pressed
- Pin = voltage level depending on key pressed
- Use 1% resistors (5% resistors + standard ADC error of 2-4% could give problems)



Power and Data on One Wire

- Line pulled low to send data
  - Can be bidirectional with input and transistor drive at both, but watch line is high long enough to drive slave



University of Strathclyde Glasgow