Orton peripheral system manual

Index

Disclaimer

The information in this document is provided purely as a record of the construction of one of my computer projects. I provide no assurance or guarantee whatsoever as to the accuracy, safety or originality of the contents of this document. I provide no warranty whatsoever as to the suitability for any purpose of the contents of this document. I accept no responsibility whatsoever for any deaths, injuries or losses resulting from the use of the information contained in this document. I have no claims to most of the technology used in my projects or described in this document, indeed, it must be assumed that I have used much copyrighted and/or patented material. But since I do what I do for purely recreational, educational or personal instructional purposes, I do not believe that I am breaking the law. It is up to the individual using the information in this document to determine whether, and to ensure that, their use of the information I provide is both legal and safe. This is supposed to be fun.

Karen Orton 2018

Chapter 1 Overview of the peripheral system

During the 1980s my then SC/MP computer found application as Eddie my house-board computer (followers of 'The Hitchhiker's Guide to the Galaxy' will understand this choice of name). Eddie was a doorbell, a clock, a burglar alarm, and a remote control for internal and external lighting. Eddie could be programmed to turn the latter on and off at preset times. Eddie's control program was written in NIBL Basic and ran in about 8kBytes of RAM. Communication was by a radio handset (me-to-Eddie) and a voice synthesiser (Eddie-to-me). While I have no plans to automate my home at the present time, I decided to implement a similar set of peripherals for my new SC/MP computer.

The peripheral system is a simple, serially controlled card frame that allows the Orton SC/MP computer to interface with a number of peripherals. The card frame has two connectors - a 5V power connector and a 9 way male D connector for serial communication. The latter is wired DTE and usually requires a cross-over (null modem) cable for attachment to a computer. 5V power may be commoned with the computer power so that both may run off a single 5V power supply. The serial port expects communication at 1200 baud, 8 data bits, no parity and 1 stop bit. RTS/CTS flow control is present and supported.

The card frame common equipment is a simple RS232 to 5VCMOS level translator. The 5V logic signals, along with 5V power, are made available on a ten conductor ribbon bus that visits every card in the frame. The signals on this bus are as follows:

Conductor	Signal	Direction	Float state
1	GND	-	-
2	CTS	Computer -> Peripherals	-
3	GND	-	-
4	RTS	Peripherals -> Computer	+3V
5	GND	-	-
6	RxD	Computer -> Peripherals	-
7	+5V	-	-
8	TxD	Peripherals -> Computer	+3V
9	+5V	-	-
10	+5V	-	-

The CTS and RxD lines simply fan out to all peripherals. The RTS and TxD lines have pull-ups to +3V and are expected to be driven by open collector drivers in 'wired-OR' fashion. An open collector driver can be approximated by a Schottky diode on the output of a standard bipolar driver.

Note that the sense of the RTS line is reversed. Normally, a low state on the RTS line ENABLES remote computer sending. Inversion is used so that any peripheral can INHIBIT remote computer sending by pulling the RTS line to the low state. CTS

must be monitored by peripherals so that responses are not sent to the computer before it is ready.

Dialogue between computer and peripherals is by simple lines of text. There is no hardware address decoding in the card frame common equipment. Instead, peripheral selection is achieved by dedicating the first character of each line sent by the computer as a command. Peripherals must therefore examine the first character of each line sent by the computer to determine whether they need to respond.

Peripheral cards frequently respond to more than one command, the typical situation requiring one command for setting internal parameter(s) and another for interrogating them. This separation of setting and interrogation simplifies the firmware in the peripheral microcontrollers. Currently, the following commands are defined:

Command character	Command
A	Send allophones to speech synthesiser
B	Interrogate remote control receiver key
C	Set time of day clock
D	Interrogate time of day clock
E	Set non-volatile memory
F	Interrogate non-volatile memory
G	Set digital outputs
H	Interrogate digital inputs and outputs

A peripheral card may be host to several logical peripherals, in which event the resident microcontroller will respond to several commands. The above commands are implemented by three peripheral cards:

Card	Logical peripheral(s)
1	Allophone speech synthesiser
2	Remote control receiver / time of day clock / non-volatile memory
3	Digital input / digital output

PARAMETERS are passed from computer to peripheral as numeric values separated by spaces. Alternatively, the command line might be a single block of text which may include spaces. These follow immediately after the command character. Command lines from the computer are terminated by a Carriage Return character (0x0D) and a Line Feed character (0x0A).

READINGs are returned from peripheral to computer as numeric values separated by commas. Alternatively, a response might be a single block of text which may include spaces. Peripherals do not begin their response lines with a command character and response lines are terminated by a Carriage Return character (0x0D)

only. The use of commas between readings ensures compatibility with the BASIC INPUT statement.

Physically, each peripheral card is 64mm by 104mm, the ten pin bus connector having a defined position and orientation to allow easy dressing of the bus cable. Also defined is the position of a 3mm red activity LED, which is always located next to the bus connector.

Chapter 2 Detailed peripheral descriptions

Command A: Send allophones to speech synthesiser

The speech synthesiser is an emulation of the General Instruments SP0256-AL2 IC. The synthesised speech is not as clear as that of the original IC. This is because the original chip had a vocal tract model that could glide smoothly between vowel sounds, whereas the emulation moves abruptly between sampled sounds. A comb filter has been added to the emulation to mitigate some of the abruptness.

The command line carries a single block of text, the characters of which a are interpreted as SP0256-AL2 allophones:

PA1	@
PA2	A
PA3	B
PA4	C
PA5	D
OY	E
AY	F
EH	G
KK3	H
PP	I
JH	J
NN1	K
IH	L
TT2	M
RR1	N
AX	O
MM	P
TT1	Q
DH1	R
IY	S
EY	T
DD1	U
UW1	V
AO	W
AA	X
YY2	Y
AE	Z
HH1	[
BB1	\
TH	]
UH	^
UW2	_

AW	` (or space)
DD2	a
GG3	b
VV	c
GG1	d
SH	e
ZH	f
RR2	g
FF	h
KK2	i
KK1	j
ZZ	k
NG	l
LL	m
WW	n
XR	o
WH	p
YY1	q
CH	r
ER1	s
ER2	t
OW	u
DH2	v
SS	w
NN2	x
HH2	y
OR	z
AR	{
YR	|
GG2	}
EL	~
BB2	?

The reader is referred to SP0256-AL2 documentation for further information on the use of allophones. No means is provided by which the computer can determine when an utterance has completed. It is expected that dialogue with the user provides these cues. To allow for phrases of realistic complexity, the speech synthesiser includes a FIFO buffer which can hold up to eighty allophones.

Command B: Interrogate remote control receiver

There is a single reading which reports a key press:

Reading	Meaning
0	No key press received
1	'1' key pressed
2	'2' key pressed
3	'3' key pressed
4	'4' key pressed
5	'5' key pressed
6	'6' key pressed
7	'7' key pressed
8	'8' key pressed
9	'9' key pressed
10	'0' key pressed
11	'CMD' key pressed
12	'END' key pressed

Note that the received key presses are 'debounced' to avoid spurious reports. Also, each key press is reported only once, thereby avoiding the need for the computer to identify the beginning and the end of each key press.

Command C: Set time of day clock

Three parameters set the clock as follows:

Parameter	Valid range	Meaning
1	0..23	Hours
2	0..59	Minutes
3	0..59	Seconds

Command D: Interrogate time of day clock

Three readings report the current time in the same order as for setting.

Command E: Set non-volatile data peripheral

The command line carries a single block of text of up to 127 characters, which is recorded into a flash memory such that it can be re-read following a power cycle.

Command F: Interrogate non-volatile data peripheral

The response is a single block of text, read from the non-volatile memory.

Command G: Set digital outputs

The digital I/O peripheral allows four bits of isolated digital output to be controlled and four bits of isolated digital input to be sensed. The expectation is that a separately powered external system will interface to these, the isolation ensuring that no current can flow from this external system into the peripheral system or the computer. Four parameters define new settings for the digital outputs:

Parameter	Meaning
1	New setting for digital output 1
2	New setting for digital output 2
3	New setting for digital output 3
4	New setting for digital output 4

Expected values for each parameter are either '0' (turn output off) or '1' (turn output on).

Command H: Interrogate digital inputs

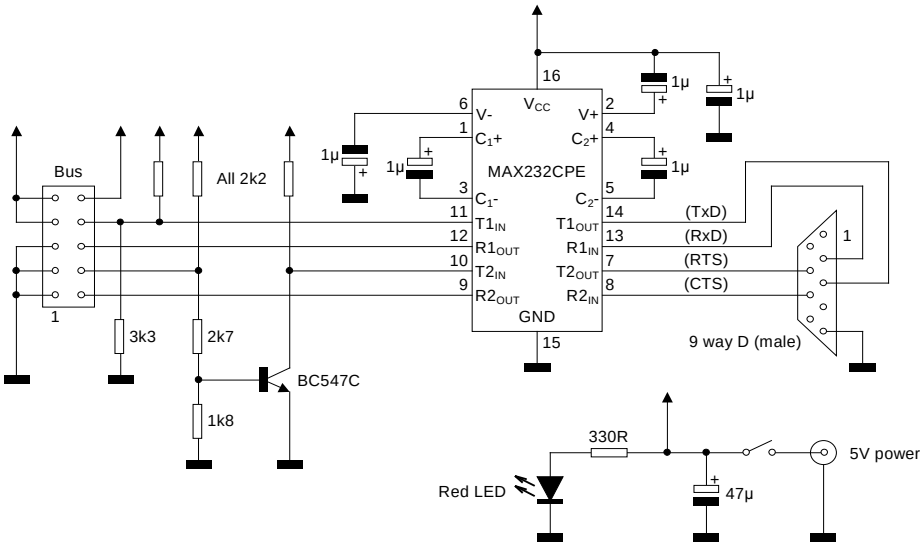
Eight readings report the current state of the digital outputs and digital inputs:

Reading	Meaning
1	Current setting of digital output 1
2	Current setting of digital output 2
3	Current setting of digital output 3
4	Current setting of digital output 4
5	Sensed state of digital input 1
6	Sensed state of digital input 2
7	Sensed state of digital input 3
8	Sensed state of digital input 4

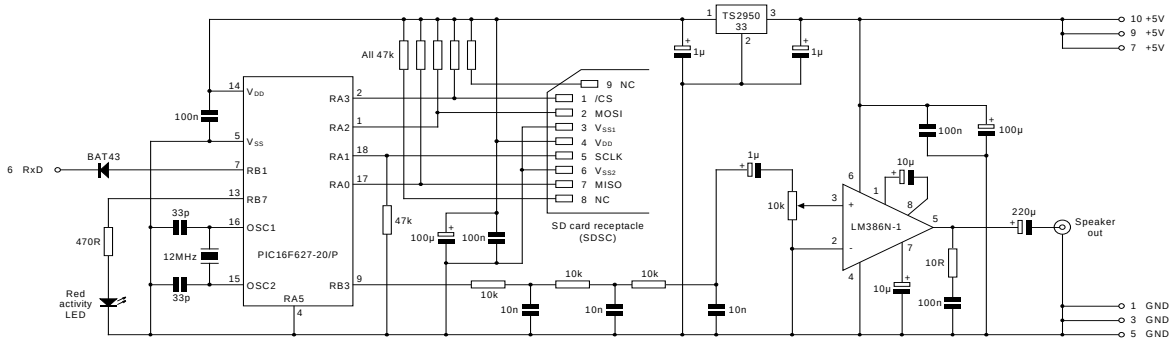
Reported values for each reading are '0' (input/output off) or '1' (input/output on).

Chapter 3 Hardware

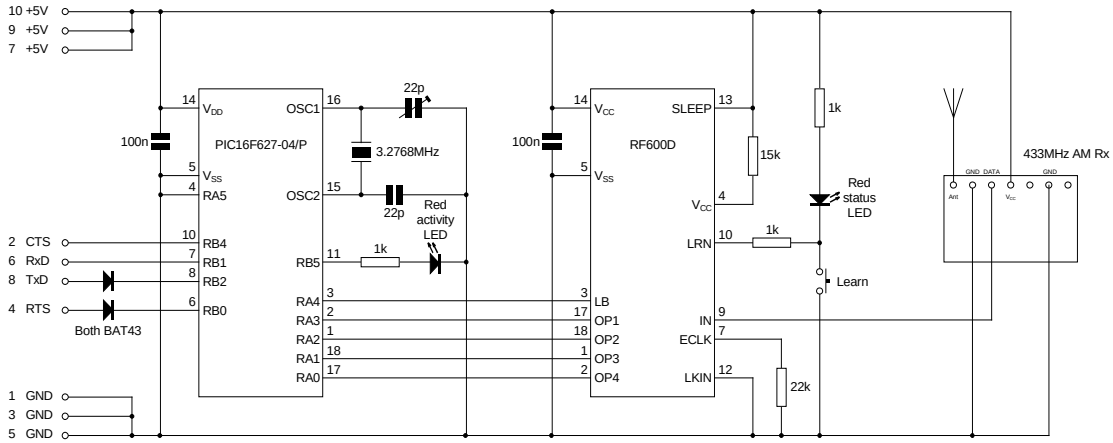
Card frame common equipment



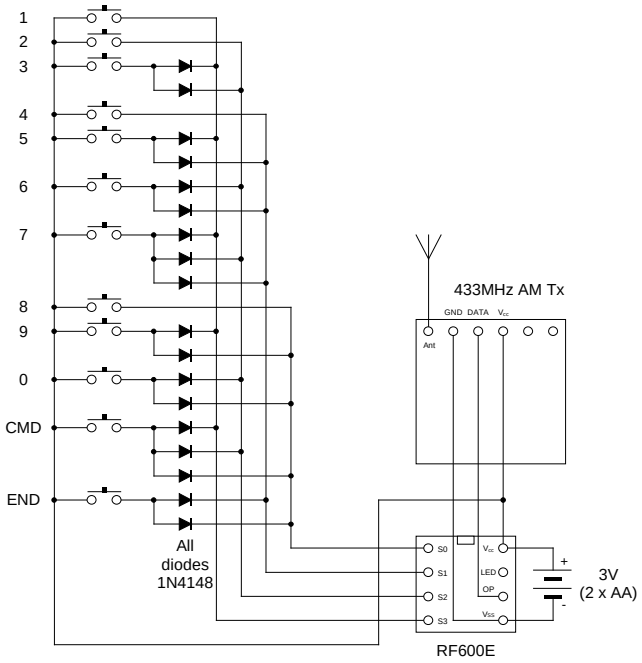
Speech synthesiser



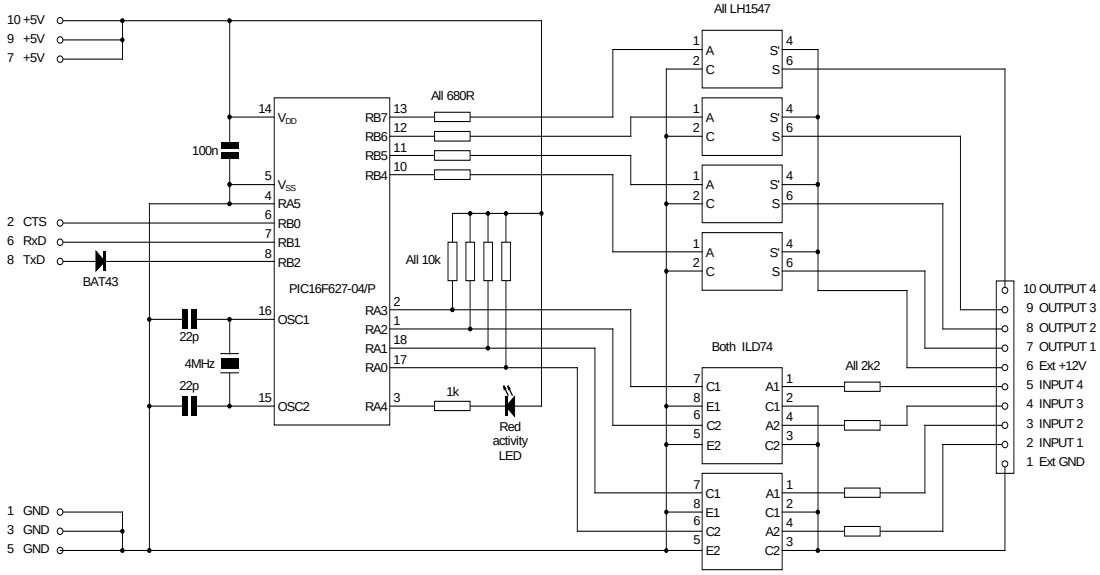
Remote control receiver, time of day clock and non-volatile memory



Remote handset



Digital I/O


All LH1547
All 680R
All 10k
All 2k2
Both ILD74
PIC16F627-04/P
100n
22p
4MHz
22p
BAT43
1k
Red activity LED
10 +5V
9 +5V
7 +5V
2 CTS
6 RxD
8 TxD
1 GND
3 GND
5 GND
10 OUTPUT 4
9 OUTPUT 3
8 OUTPUT 2
7 OUTPUT 1
6 Ext +12V
5 INPUT 4
4 INPUT 3
3 INPUT 2
2 INPUT 1
1 Ext GND

Photos

Chapter 4 Software

There are a number of supporting files associated with the peripheral system:

File	Purpose
spsyn.asm	PIC source for the speech synthesiser peripheral card
allo.bin	Binary contents of speech synthesiser SD card
remclk.asm	PIC source for the remote receiver, time of day clock and non-volatile memory peripheral card
dio.asm	PIC source for the digital I/O peripheral card