

# SMARTCARD CONTROL SYSTEM

---

## INTRODUCTION

The SmartCard Control System uses a credit-card sized SmartCard to store 'control' programs. The SmartCard uses 'EEPROM' memory which will retain the control program permanently, but can also be reprogrammed any number of times. The remainder of the system consists of two separate electronic modules - the SmartCard Programmer and the SmartCard Reader. The Programmer module is used to program the SmartCard with a sequence of commands. Once programmed, the SmartCard is then placed in the low-cost Reader module so that the program can be run.

The Reader module is designed to be 'built into' a project, whilst the Programmer module is designed to be used as a stand-alone unit. By programming two or more SmartCards it is possible for a number of programs to be run.

## CONTROLLER MODULE

The Reader module supports 6 digital output lines, 2 digital input lines and 1 'reset' input line. All output lines are 'buffered' via a Darlington driver so that output devices such as motors and buzzers can be directly attached to the controller board. The Reader can be powered by any battery pack in the range of 6 to 12V.

## PROGRAMMER MODULE

The Programmer module supports three different programming methods. Two of these methods use manual switches on the programmer unit, and so no computer is required. The third method uses a serial link to a host computer running the SmartLink software.

## PROGRAMMING MODES

### **Bit-by-Bit Mode**

This is the simplest programming method, where output sequences are entered into the SmartCard memory line by line via the on-board switches. This programming mode is based on the methods used with the TEP Bit-by-Bit Sequential Controller, and so users familiar with this device will find the programming method very similar. The Bit-by-Bit Comparison Study File explains the differences between the two systems.

### **Smart Basic Mode**

This programming mode uses six simple 'BASIC' style commands to form a control program up to 100 lines long. The commands are selected via push switches on the programmer board, and the program lines are displayed on a text Liquid Crystal Display (LCD), so that no computer is required. The 'if' command allows conditional commands to be used, and so 'feedback' from the digital inputs can be included within the control program.

### **Smart Link Mode**

This programming mode uses the same six BASIC commands as the Smart Basic Mode, but the program is written on a computer and then downloaded to the Programmer. Programs present on the SmartCard can also be 'read' for editing, and programs entered manually using the Smart Basic programming method can later be 'read' and printed out. The programmer connects to a serial port on the computer via a standard serial cable. The software also includes a 'simulation' feature, so programs can be 'tried out' via an on-screen simulation.

The remainder of this study file is split into five sections. You should read the first two sections, and then the section explaining the programming mode that you wish to use.

- **The SmartCard Reader**  
A description of the Reader module, describing it's main components.
- **The SmartCard Programmer**  
A description of the Programmer module, describing it's main components.
- **Using the Bit-by-Bit Programming Mode**  
How to program the SmartCard using the Bit-by-Bit programming mode.
- **Using the Smart Basic Programming Mode**  
How to program the SmartCard using the Smart Basic programming mode.
- **Using the Smart Link Programming Mode**  
How to program the SmartCard using the computer software and a serial link to the Programmer module.

## THE SMARTCARD READER

The Reader module 'runs' the program saved on the SmartCard. You cannot use the Reader to program the SmartCard, this must be carried out using the Programmer Module. Using the Reader is very straight forward - simply connect the battery and then insert the card. The program stored on the SmartCard will start running automatically. To re-start the program press the 'reset' switch.

### **SmartCard Socket**

This is where the SmartCard is inserted. The SmartCard must be inserted with the contacts facing down.

### **PIC Microcontroller**

This is the 'brain' of the Reader, which reads the commands saved on the SmartCard and then carries them out.

### **5V Regulator**

This 'regulates' a steady 5V supply for the microcontroller and SmartCard. The regulator will not operate under 6V, and so the supply voltage must be at least 6V for correct operation. Any supply voltage between 6 and 12V may be used.

### **Darlington Driver IC**

This integrated circuit (IC) contains seven Darlington transistors as well as protection diodes, and is used instead of placing a single darlington transistor (like the BCX38B) and diode on each output line. Each open collector output line can therefore sink 550 mA of electric current. The output voltage will be at the supply voltage (6-12V).

### **Output LEDs**

These red LEDs show the condition of the six outputs. These LEDs will not function if you remove the Darlington driver chip from the pcb.

### **Input LEDs & DIP switch**

The yellow LEDs show the condition of the two input lines. The DIP switch can be used to switch the lines on or off, this is useful when testing programs. If you connect your own switch to the input lines you must ensure that the DIP switch is placed in the off position, or the DIP switch will 'over-ride' the remote switch, causing the input line to be permanently on.

### **Reset Switch**

This resets the module, so that the control program starts running from the first line again.

### **Card Detect LED**

This green LED lights when a program is being run. The LED flashes to show an error condition- this will normally be caused by a SmartCard being inserted the wrong way around. If this happens remove the card and then re-insert it correctly.

### **EEPROM socket**

This socket can be used to hold an EEPROM chip instead of using the SmartCard. See the 'Using an EEPROM chip' study file for more details.

## THE SMARTCARD PROGRAMMER

The Programmer module is used to program the SmartCard. You cannot 'run' programs on the Programmer, this must be carried out using the Reader module.

### **SmartCard Socket**

This is where the SmartCard is inserted. The SmartCard must be inserted with the contacts facing down.

### **PIC Microcontroller**

This is the 'brain' of the Reader, which 'interprets' the switch pushes and then programs the commands onto the SmartCard.

### **5V Regulator**

This 'regulates' a steady 5V supply for the microcontroller and SmartCard. The regulator will not operate under 6V, and so the supply voltage must be at least 6V for correct operation. A 9V PP3 battery is recommended for use with the Programmer module.

### **Mode/Program switch**

These two switches are used to select the programming mode.

### **Push switches**

These switches are used to enter the program commands.

### **Reset Switch**

This resets the Programmer module. It is good practice to press the reset switch after connecting the battery to ensure that the module has reset correctly.

### **Card Detect LED**

This green LED lights when a SmartCard has been detected. The LED flashes to show an error condition- this will normally be caused by a SmartCard being inserted the wrong way around. If this happens remove the card and then re-insert it correctly

### **Output LEDs**

These red LEDs show the condition of the six outputs when programming in 'Bit-by-Bit' mode. They are not used with the other programming modes.

### **Liquid Crystal Display (LCD) (*optional*)**

This display is used with the 'Smart Basic' programming mode to display the commands as they are entered.

### **Serial Socket**

This is where the serial cable for the 'Smart Link' programming mode is connected.

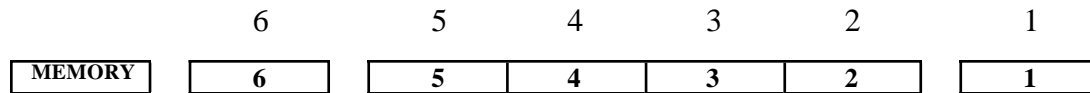
### **EEPROM socket**

This socket can be used to hold an EEPROM chip instead of using the SmartCard. See the 'Using an EEPROM chip' study file for more details.

BIT-BY-BIT PROGRAMMING MODE

**Using the Programmer Module**

The Bit-by-Bit programming mode uses the seven push switches and 6 red LEDs on the Programmer. The labels printed above and below the switches can be ignored as they are descriptions used with the Smart Basic programming mode. For the Bit-by-Bit mode the switches should be regarded as shown below.



**Watch your fingers!**

Always rest the programmer on a flat surface when it is in use. If you 'hold' the Programmer in the air your fingers will probably touch the electrical contacts on the rear of the pcb. This will not cause you or the module any harm, but the moisture on your fingers can affect the circuit so that the electronic signals will become corrupted, and this will cause an error. If an error occurs the green LED will start flashing and you must then reset the module using the Reset switch.

**Getting Started**

As an example, we shall program the SmartCard with the program below, which will cause outputs 1 and 2 to flash on and off alternately every three seconds. The program is six lines long (when using a run speed of 1 line per second).

```
Line 1 ; switch on output 1 for 1 second
Line 2 ; switch on output 1 for 1 second
Line 3 ; switch on output 1 for 1 second
Line 4 ; switch on output 2 for 1 second
Line 5 ; switch on output 2 for 1 second
Line 6 ; switch on output 2 for 1 second
```

Ensure the 'Mode' and 'Program' switches are in the 'off' position and then connect the battery to the Programmer module. It is good practice to press the 'Reset' switch after connecting the battery to ensure that the Programmer has reset correctly.

Insert a Smart Card (with the contacts down). The green 'Card Detect' LED should light to show that the Programmer has detected the SmartCard. If the LED starts flashing an error has occurred with the card - this is normally due to the card being inserted upside down. Remove the card and then re-insert it correctly.

Switch the 'Mode' switch to 'Bit-by-Bit' mode. The LEDs may flicker briefly as this happens - this is a characteristic of the Programmer module and can be safely ignored.

Now switch the 'Program' switch to the 'Program On' position. The SmartCard is now ready to be programmed.

To program a line, switch on the appropriate output LEDs by pressing the corresponding switch. In this case, press switch 1 so output LED 1 lights. The switches work on a 'latching' principle - press once to switch on, press again to switch off. When the first LED is lit correctly press the 'memory' switch to save the line. All the LEDs will flash to show that line has been saved.

Repeat this process for the remaining lines. When all six lines have been entered switch the 'Program' switch to the 'Program Off' position. This step is **essential** - if you remove the SmartCard without switching the 'Program' switch off the program will **not** run correctly.

It is now necessary to program the 'Run Speed' onto the SmartCard.

### **Setting the Run Speed**

Hold down the 'Memory' switch and move the 'Program' switch to the 'Program On' position. Sample program 1 requires a run speed of 1 second per line. To select this run speed press input switch 3 so that LED3 lights (the run speed is shown at the bottom right of the LED). If you wish to change the setting simply press a different switch.

When you have selected the correct setting press the 'memory' switch. The LED will now start flashing to indicate that that speed setting has been saved. Switch the 'Program' switch back to the 'Program Off' position.

Remove the SmartCard from the Programmer module.

### **Running the Program**

Connect the battery to the Controller module. Make sure that the 'Pause Input' switches are off and then insert the SmartCard.

The green 'Card Detect' LED will light, and the program will start to run immediately. If the green LED starts flashing an error has occurred - check that the SmartCard is inserted the correct way up.

If you wish to start the program again press the 'Reset' switch. To pause the program switch either of the two 'Pause Inputs' on. The yellow LEDs will light to show the pause condition. The program will stop at the current line until the pause switch is released.

The program will continue running until the card is removed (or the battery is removed).

## SMART BASIC PROGRAMMING MODE

This programming mode is used to program the SmartCard with a series of commands. The program can include 'conditional commands' (commands that are only carried out when certain input conditions are true), and loops.

This mode supports 6 digital outputs and 2 digital inputs.

## PROGRAM STRUCTURE

The SmartCard supports a control program up to 100 lines long (line numbers 00 to 99). When the program is run, each command from line 00 onwards is carried out sequentially. However, the **goto** command can be used to cause the program flow to jump to a different line. Therefore, by using the **goto** command, permanent looping programs can be created.

## SMARTBASIC COMMANDS

The SmartBasic language used to program the SmartCard consists of 6 commands. These commands are as follows:

- **high *pin***  
Switches on one of the six outputs. *pin* must be the output number 1 to 6
- **low *pin***  
Switches off one of the six outputs. *pin* must be the output number 1 to 6
- **out *value***  
Switches all the outputs on/off simultaneously. *value* must be a number between 00 and 63
- **wait *time***  
Pause the program for a length of time. *time* is the pause length (00 to 99) in 1/10 second units.  
Therefore the command wait 10 creates a pause one second long.
- **goto *line***  
Jump to a specific program line. *line* is the program line to jump to (00 to 99)
- **if *xy*...**  
The if command can be added before any of the above commands to give a conditional statement, *xy* being the condition. If the condition is not true the rest of that line is ignored.

For users familiar with PBasic (Basic Stamp or 'TEP PLC Chip') or traditional 'Logo' type control software, the following command comparison table may be of use:

SmartBasic Command	PBasic	'Traditional' Command
high	high	switch on
low	low	switch off
out	pins =	output, outport, out
wait	pause	wait
goto	goto	-
if...	if pin... then...	if input... then...

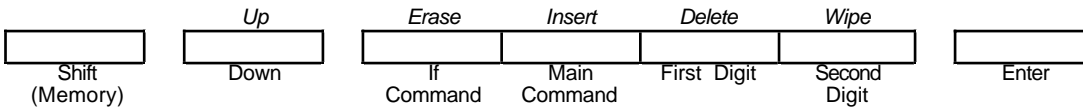
### USING THE PROGRAMMER MODULE

When a SmartCard is inserted into the Programmer Module the current program stored on the card is automatically read. It is then possible to edit this program, or simply erase the program and start again. Two program lines are shown on the LCD display at any one time- the 'active' line for editing is always the bottom line.

Typical LCD reading:

00	high 1
01	if -H goto 00

Programming uses seven push switches - four 'command' switches, one 'enter' switch, one 'up/down' switch and one 'shift' switch, which are arranged as shown.



### COMMAND SWITCHES

These four switches are used to edit the command line. Each switch controls part of the command line as follows:

If Command	Main Command	First Digit	Second Digit
(none)	(none)	0	0
if -L	high	1	1
if -H	low	2	2
if L-	out	3	3
if H-	wait	4	4
if LL	goto	5	5
if LH		6	6
if HL		7	7
if HH		8	8
		9	9

As each switch is pressed it 'cycles' through each of the options as shown in the table. Therefore to change a command you press each switch in turn until the command you want to use is displayed on the LCD. When the whole line is correct press the 'enter' button to save the line and move down to the next line.

The first digit and second digit are 'linked' to the main command. This means that you cannot enter numbers until a main command is displayed. The switches are also 'intelligent' and will only allow valid numbers - for instance **out 64** can not be selected as the highest possible out command is **out 63**.

To move down the lines without saving any changes press the 'down' switch rather than the 'enter' switch. This is useful if you want to edit a program as it allows you to move down the program lines without making any changes.

### SHIFT SWITCH

The shift switch operates in exactly the same way as the 'Shift' key on a computer keyboard, allocating a different function to five of the other buttons. These five functions are:

- Up**            Move up one line.
- Erase**        Erases the whole program for a fresh start.
- Insert**        Insert a new line. All lines below the current line are shifted down one, and a new line is inserted. Line 99 is lost permanently.
- Delete**        Delete the current line. All lines below the current line are shifted up one.
- Wipe**          Wipe all the commands on the current line

The Erase, Insert and Delete functions display a 'confirm' message before the operation is carried out to prevent accidental mistakes. The 'Shift' switch has no effect on the 'Enter' switch.

If you use the 'Insert Line' or 'Delete Line' functions remember that you may have to alter any 'goto' commands, as the line numbers will have changed after the function is carried out.

### Watch your fingers!

Always rest the programmer on a flat surface when it is in use. If you 'hold' the Programmer in the air your fingers will probably touch the electrical contacts on the rear of the pcb. This will not cause you or the module any harm, but the moisture on your fingers can affect the circuit so that the electronic signals will become corrupted, and this will cause an error. If an error message appears on the LCD you must reset the module using the Reset switch.

GETTING STARTED

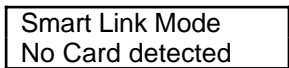
As an example, we shall program the SmartCard with sample program 1, which will cause output 1 to flash on and off every second. This program is five lines long (and so uses program lines 00 to 04).

**Sample program 1**

This program cause output 1 to flash on and off every second.

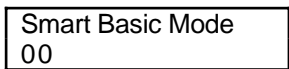
```
00      high 1          ; switch on output 1
01      wait 10         ; wait 1 second
02      low 1           ; switch off output 1
03      wait 10        ; wait 1 second
04      goto 00        ; loop back to line 00
```

Ensure the 'Mode' and 'Program' switches are in the 'off' position and then connect the battery to the Programmer module. It is good practice to press the 'Reset' switch after first connecting the battery to ensure that the Programmer has reset correctly.



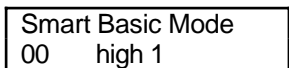
The LCD message should appear like this. Insert the Smart Card (with the contacts down) and the bottom line will change to 'Card detected'. If an error message appears remove the card and then re-insert it the correct way around.

Now switch the 'Program' switch to the 'Program On' position. The LCD will display the message 'Smart Basic Mode' followed by line 00. If you have not programmed the card before this line will be blank.



When writing a new program it is a good idea to erase the SmartCard memory to enable a 'fresh start'. Therefore hold 'Shift' down and press the 'If Command' switch. When the 'confirm' message appears press 'Enter' and then wait for the erase to finish.

The first line of the program is **high 1**, and so press 'Main Command' until the word 'high' appears. Then press 'First Digit' until the number 1 appears.



Press 'Enter' to save the line. The line will be saved, and the display will move down one line. The second line is **wait 10**, and so press 'Main Command' until the word 'wait' appears. Then use the two digit switches to set the time to '10'.

00	high 1
01	wait 10

Press 'Enter' to save the second line, and then complete lines 02, 03 and 04 in a similar manner. When you have completed the program use the 'Down' (and 'Up') switches to move up and down the lines so that you can check your program

Switch the 'Program' switch to the 'off' position and then remove the SmartCard from the Programmer Module.

### RUNNING THE PROGRAM

Connect the battery to the Controller module. Make sure that the 'Input' switches are off and then insert the SmartCard.

The green 'Card Detect' LED will light, and the program will start to run immediately. If the green LED starts flashing an error has occurred - check that the SmartCard is inserted the correct way up.

If you wish to start the program again press the 'Reset' switch.

Normally when the program has finished the green LED will go out. However, with this example, we have created a 'looping' program that will continue running until the card is removed (or the battery is removed).

*You have now learnt how to program and run programs using the high, low, wait and goto commands. The rest of this section explains how to use the other SmartBasic Commands - **out** and **if**. You should try programming and running all the other sample programs listed to ensure you understand how each of the commands work.*

### SWITCHING ALL OF THE OUTPUTS USING THE OUT COMMAND

The **out** command controls all the outputs, switching all outputs on/off simultaneously. This command can be used instead of several **high/low** commands. Each output is allocated a value as shown in the following table.

<b>Output</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
<b>Value</b>	<b>3 2</b>	<b>1 6</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>

The out value is the sum of the value of each output you want to be high (switched on). Therefore the command to switch on outputs 1, 3 & 5 would be **out 21** (as  $21 = 1+4+16$ ). All the other outputs will be switched off. The maximum out value is **out 63**, which will cause all the outputs to be switched on.

**Sample program 2**

With this program all the outputs flash on and off every 2 seconds.

```
00      out  63      ; switch on all the outputs
01      wait 20     ; wait 2 seconds
02      out  00     ; switch off all the outputs
03      wait 20     ; wait 2 seconds
04      goto 00     ; loop back to line 00
```

USING INPUTS WITH THE IF... COMMAND

The **if...** command can be added in front of any of the other command to create a conditional command. This means that the main command is only carried out if the input condition is true. If the input condition is not true the main command is ignored, and program flow moves to the next line.

There are 8 different **if** command possibilities, which use the following codes for abbreviation:

H = high (on) , L = low (off), - = don't care.

Command	Description	True if...	or...
if -L	if input 1 is low...	$\infty$ 2 1	$\neq 0$ 2 1
if -H	if input 1 is high...	$\neq 0$ 2 1	$\neq$ 2 1
if L-	if input 2 is low...	$\infty$ 2 1	$\neq 0$ 2 1
if H-	if input 2 is high...	$\neq 0$ 2 1	$\neq$ 2 1
if LL	if input 2 and input 1 are both low...	$\infty$ 2 1	
if LH	if input 2 is low and input 1 is high..	$\neq 0$ 2 1	
if HL	if input 2 is high and input 1 is low...	$\neq 0$ 2 1	
if HH	if input 2 and input 1 are both high...	$\neq$ 2 1	

**Sample program 3**

This program 'echoes' the inputs condition on outputs 1 and 2 by use of the **out** command.

```
00  if LL out 00      ; both LEDs off
01  if LH out 01      ; one LED on
02  if HL out 02      ; the other LED on
03  if HH out 03      ; both LEDs on
04      goto 00      ; loop back to line 00
```

**Sample Program 4**

This program switches on a different output according to which input is pressed.

```

00  if -H goto 08      ; if input 1 is on goto line 08
01  if H- goto 15     ; if input 2 is on goto line 15
02          goto 00   ; loop back to line 00

08      high 1       ; switch on output 1
09      wait 30      ; wait 3 seconds
10      low 1        ; switch off output 1
11      goto 00     ; loop back to line 00

15      high 2       ; switch on output 2
16      wait 30      ; wait 3 seconds
17      low 2        ; switch off output 2
18      goto 00     ; loop back to line 00

```

*Note that this program leaves a number of 'empty' lines between the three sections. This can be a useful programming method if you think you may want to add extra lines at a later date. However you must ensure that the program cannot jump to one of the unprogrammed lines, as this will cause your program to stop unexpectedly.*

**Sample program 5**

*This program waits until input 1 is switched on, and then switches on output 3 for as long as input 1 is kept high. This demonstrates how 'jumping' to the same line can create a 'wait until input changes' type of command.*

```

00  if -L goto 00     ; if input 1 is off then stay here
01      high 3       ; switch on output pin 3
02  if -H goto 02     ; stay here until input 1 is off again
03      low 3        ; switch off output pin 3
04      goto 00     ; loop back to line 00

```

**SMART LINK PROGRAMMING MODE**

This programming mode is used to program the SmartCard via a computer. The software uses the same commands as the Smart Basic Programming Mode, and so you should read through that section now if you have not already done so. No explanation of the Smart Basic commands used will be repeated in this section.

For details on how to install the software see the Software Installation section.

**Connecting the Programmer to the Computer**

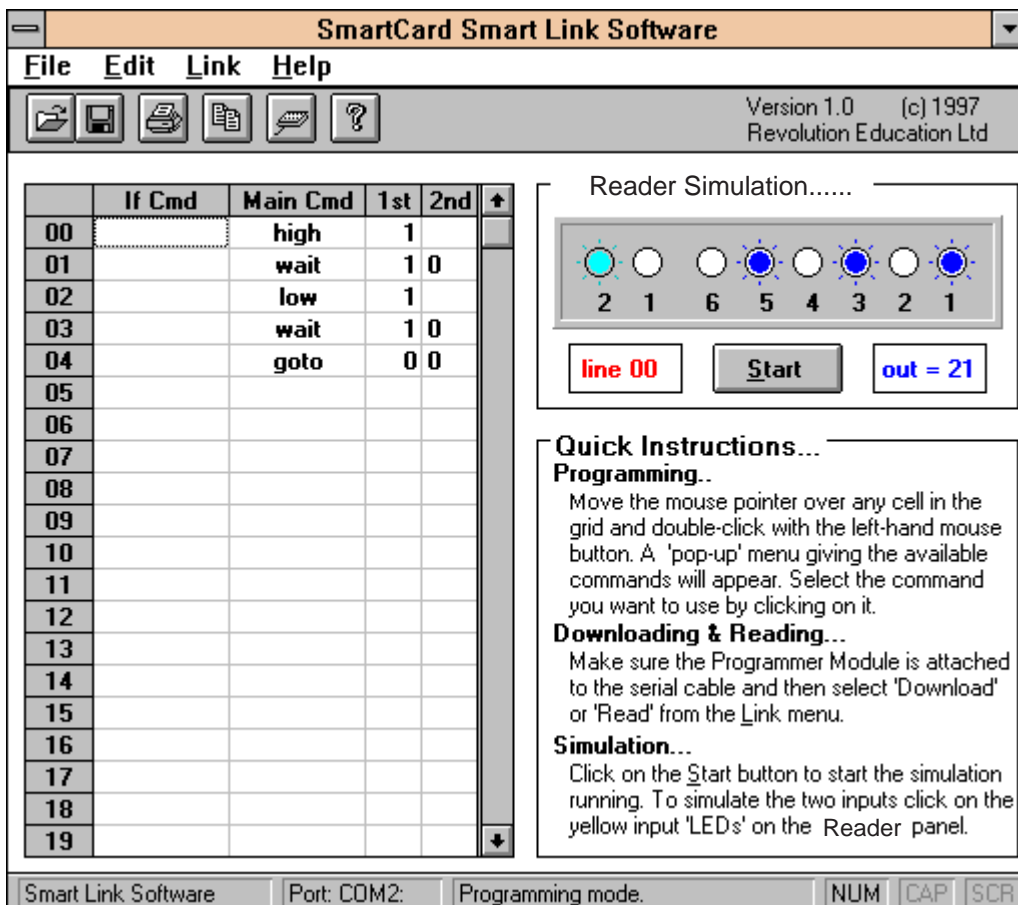
- **PC (9 pin socket)**  
Connect the serial cable from the programmer to the serial port on the computer.
- **PC (25 pin socket)**  
Connect the serial cable to the programmer. Use a 9 to 25 way 'mouse' adapter to connect the other end of the cable to the computer.
- **Mac**  
Connect the serial cable from the Programmer to the 'Modem' or 'Printer' port on the computer.

Connect a battery to the programmer and insert a SmartCard. Check that the green Card Detect LED is lit - if it is flashing an error has occurred, normally due to the SmartCard being inserted upside down. Remove the card and then re-insert it the correct way around.

Start up the software by clicking on the SmartCard Smart Link icon.

**USING THE SOFTWARE**

When you start the software the following window will appear:



**Menu Bar & Toolbar**

These contains the usual menu options and 'shortcut' toolbar buttons.

**Programming Grid**

The grid contains the actual program code. You can only see 20 of the program lines at once, and so the scroll-bar at the side of the grid must be used to move up and down the program.

**Simulation Window**

This window shows the on-screen simulation of the Reader Module for 'trying out' programs.

**Status Bar**

The status bar gives comments on how the software is configured.

PROGRAMMING

The programming grid operates in a very similar manner to the manual SmartBasic Programming mode. Each program line is split into four sections - if command, main command, first digit and second digit. To edit a line simply double-click on the part of the grid that you would like to change, and a pop-up menu of the available options will appear. Select the option you require by clicking on it

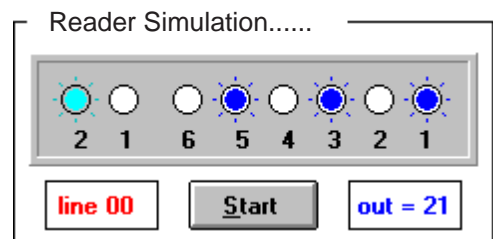
Note that the first and second digits are 'linked' to the main command, and so you will not be able to select a number if no main command has been selected. The computer will beep if you try to select an unavailable digit in this way.

DOWNLOADING

To download a program, make sure the Programmer is attached, and then click on the 'Download' button on the toolbar. The program will then be downloaded to the SmartCard. If an 'error' message appears check that the cable and battery are connected to the Programmer, that the SmartCard is inserted correctly, and that both the 'Mode' and 'Program' switches are in the 'off' position on the Programmer. Also check that you have selected the correct serial port via the 'Link' menu (The serial port currently selected is shown on the Status Bar)

READ

To 'read' a program from the SmartCard make sure that the Programmer is attached and then select 'Read' from the 'Link' menu. The program will then be read from the SmartCard. If an 'error' message appears check that the cable and battery are connected, that the SmartCard is inserted correctly, and that both the 'Mode' and 'Program' switches are in the 'off' position on the Programmer. Also check that you have selected the correct serial port via the 'Link' menu. (The serial port currently selected is shown on the Status Bar)



## PROGRAM SIMULATION

The simulation window allows you to 'try-out' programs without downloading them to the SmartCard. This allows you to rapidly experiment with different programs.

To start the simulation running click on the 'Start' button. The program will then run on-screen, starting from line 00. When the program reaches an unprogrammed line it will stop automatically.

A 'looping' program can be stopped at any time by clicking on the 'Stop' button or by pressing the <Esc> or <Spacebar> keys on the keyboard.

To simulate an input click on the appropriate input LED to switch it on ("high") or off ("low")

The speed the simulation runs at depends on your computer type. However no simulation will run as fast as the real Reader module, which can easily carry out all 100 program lines in under a second! If the simulation is running too fast to understand, you can add an extra time delay between each line to 'slow it down'. To do this select 'Simulation Speed' from the 'Edi' menu and select a delay between 0 to 5 seconds long..

## TIP

When a program is not being run you can click on the 'output' LEDs to switch them on and off. This alters the 'out' value shown beneath the LEDs. Therefore you can quickly calculate the **out** value to be used in a program by simply clicking on the LEDs until the correct output combination is shown.

You cannot click on the output LEDs when a simulation is running (although you can still click on the input LEDs to change their value).

## SOFTWARE MENUS

Note that most menu commands also have a shortcut button on the toolbar.

### File Menu

New...	- start a new program
Open...	- open a file saved on disk
Save...	- save the current program
Save As...	- save the current program with a new name
Print...	- print the current program
Printer Setup...	- select the printer configuration
Exit	- exit the software

**Edit Menu**

- Insert Line... - insert a new program line
- Delete Line... - delete the current program line
- Simulation Speed - alter the time delay between the simulation steps
- Titles... - add a title for printed / copied programs
- Copy - copy the program to the Windows clipboard

**Link Menu**

- Port... - select the serial port to use
- Read - read a program from the SmartCard
- Download - download a program to the SmartCard

**Help Menu**

- Help Contents - start the Help File
- Register - registration details
- About... - version details, programmer details and copyright notice

SMART BASIC COMMANDS SUMMARY

**Main Commands**

Command	Description
high	Switches an output high ("on")
low	Switches an output low ("off")
out	Switches all the outputs high or low ("on/off") <i>See table below for further details.</i>
wait	Pauses the program for a length of time. wait 01 = wait for 1/10th of a second wait 10 = wait for 1 second
goto	Jump to a different program line.

**'out' Command**

The **out** command switches all the outputs on and off simultaneously. The value of the out command (00 to 63) is calculated using this table. Add the values for each output you want to be switched on.

Output Number	6	5	4	3	2	1
Value for "out" command	32	16	8	4	2	1

Example - Switch outputs 2, 4 and 6 are high, all the others are low

	≠	°	≠	°	≠	°
	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
<b>Value = 32 + 8 + 2 =</b>	<b>32</b>	-	<b>8</b>	-	<b>2</b>	-
<b>42</b>						

Therefore the command is **out 42**

**‘if’ Command**

When an ‘if’ command is placed before a ‘main’ command, the ‘main’ command is only carried out if the input condition is true.

Command	Description	True if..	or...
if -L	if input 1 is low...	∞ 2 1	≠° 2 1
if -H	if input 1 is high...	°≠ 2 1	≠≠ 2 1
if L-	if input 2 is low...	∞ 2 1	°≠ 2 1
if H-	if input 2 is high...	≠° 2 1	≠≠ 2 1
if LL	if input 2 and input 1 are both low...	∞ 2 1	
if LH	if input 2 is low and input 1 is high..	°≠ 2 1	
if HL	if input 2 is high and input 1 is low...	≠° 2 1	
if HH	if input 2 and input 1 are both high...	≠≠ 2 1	

**USING AN EEPROM CHIP INSTEAD OF A SMARTCARD**

It is possible to use an EEPROM chip to store the program instead of a SmartCard. This can be useful if you have written a set control program that does not have to be changed. The chip can be placed on the controller board permanently so that a SmartCard is not required.

**Using an EEPROM chip**

To program the EEPROM chip the Programmer Module must be equipped with the blue dot (Bit-by-Bit) microcontroller or red dot (Smart Basic-EEPROM) microcontroller. If your programmer has the green dot (Smart Basic-SmartCard) microcontroller you must purchase the Programmer EEPROM Chip Upgrade kit.

Place the EEPROM chip in the socket (making sure pin 1 is in the correct position) on the Programmer module and then connect the battery. The chip can then be programmed in an identical manner to the SmartCard by either of the three programming methods. Once programmed, remove the battery and then remove the chip. NEVER remove or insert the EEPROM chip when the battery is connected, as this could damage the chip.

When 'levering' the chip out take care not to bend the legs.

To run the program simply insert the EEPROM chip into the Reader module and then connect the battery. The reader will detect the chip and start running the program.

You should not try to use a SmartCard when there is an EEPROM chip in the socket, although doing this accidentally will not damage any of the modules. The EEPROM chip takes priority over the SmartCard, and so the SmartCard will be ignored.

#### SMARTCARD CONTROL SYSTEM COMPONENTS

- **Bit-by-Bit SmartCard Programming Set**
  - 1 Programmer Module (without LCD Display)
  - 1 Reader Module
  - 1 SmartCard
  - 1 'Special Publication' Booklet
  
- **SmartBasic SmartCard Programming Set**
  - 1 Programmer Module (with LCD Display)
  - 1 Reader Module
  - 1 SmartCard
  - 1 'Special Publication' Booklet
  
- **PC Smart Link Programming Kit**

Required for using the SmartLink software with the Programmer. (PCs with a 25 way serial socket will also require a 9 to 25 way 'serial mouse' adapter)

  - 1 PC Serial Cable
  - 1 Software Disk
  
- **Mac Smart Link Programming Kit**

Required for using the SmartLink software with the Programmer. Note that you also require 'SoftWindows 2.0', 'SoftWindows 3.0' or 'SoftWindows '95' to be able to use the Smart Link software on a Mac.

  - 1 Mac Serial Cable (8 pin mini-DIN)
  - 1 Software Disk

- **Programmer LCD Upgrade Kit**  
 Use to upgrade a 'Bit-by-Bit' programmer for 'Smart Basic' programming.  
 1 LCD display  
 1 14 pin header  
 4 LCD support posts  
 1 'Green dot' PIC Microcontroller.
- **Programmer EEPROM Chip Upgrade Kit**  
 Use to program EEPROM chips instead of the SmartCard  
 1 'Red dot' PIC Microcontroller  
 1 24LC02 EEPROM Chip  
 1 chip removal tool

The following components are also available individually:

- SmartCard
- Reader Module
- 'Special Publication' Booklet
- 24LC02 EEPROM Chip

#### UPGRADING THE PROGRAMMER MODULE

The Programmer Module is available in two different formats - with an LCD and without an LCD. If required the LCD can be added to the Programmer at a later date by purchasing the LCD upgrade kit.

You require a different PIC microcontroller (the Programmer "brain") when using the different options. There are three microcontrollers available, and these are indicated by a coloured 'dot' sticker when supplied. The microcontroller is exchanged by simply levering the existing microcontroller out of its socket and pushing the new microcontroller in place. No soldering or technical knowledge required!

Colour	Programming Modes	LCD required	Programs
Blue	Bit-by-Bit or Smart Link	No	SmartCard or EEPROM Chip
Green	Smart Basic or Smart Link	Yes	SmartCard only
Red	Smart Basic or Smart Link	Yes	EEPROM chip only

If you 'lose' the dots, or want to know the version number of the PIC microcontroller, connect the programmer to a computer and run the Smart Link software. Select 'About' from the 'Help' menu and the Programmer firmware version type and number will be displayed.

**Adding the LCD display**

This section explains how to upgrade the Bit-by-Bit Programmer module (by adding a LCD display) so that the Smart Basic programming mode can be used. Note that it is possible to purchase the Programmer module with the LCD in place, these instructions are only for those who purchased a Programmer without the LCD display and now wish to upgrade it.

**LCD Upgrade Kit Contents:**

1 14 pin header  
 1 LCD display  
 4 LCD support posts  
 1 PIC Microcontroller (green dot)

**Tools required:**

Soldering iron and solder.  
 Pair of side-cutters.  
 Small screwdriver.

1. Remove the battery from the Programmer Module.
2. Place the short end of the 14 pin header into the 'LCD' holes on the Programmer. Turn the programmer upside down and then solder the header in place.
3. Place the programmer the correct way up and press the 4 LCD support posts into their holes.
4. Carefully press the LCD onto the support posts, so that the header passes through holes 1 to 14 on the LCD (note that holes 15 and 16 on the LCD are not used). Solder the header in place, and then cut any long header pins to length with the side-cutters.
5. Touch a bare water pipe (or the metal housing at the rear of a computer) to discharge the static electricity from your body.
6. Use the small screwdriver to carefully remove the existing blue dot microcontroller from its socket.
7. Place the new green dot microcontroller in the socket, ensuring the notch on the chip is at the top (serial connector) end of the socket. Check that all 28 pins have entered the socket and not accidentally 'folded up' under the chip.
8. Connect the battery and then use a small screwdriver to adjust the contrast of the display (by adjusting the preset resistor through the hole in the pcb).

*If you wish to change back to the 'bit-by-bit' mode at a later date simply swap the two microcontroller chips. There is no need to remove the LCD, but it will display dark 'squares'. However it is possible to adjust the LCD contrast so that the squares are not visible!*

### BIT-BY-BIT MODE QUICK COMPARISON

This study file is for users familiar with the standard bit-by-bit controller who now wish to use the SmartCard system. It provides a brief summary of the changes used with this system - for more detailed instructions please see the main 'Bit-by-Bit' programming section.

#### **Programming**

The SmartCard is programmed using the Programmer module, and run using the Reader module. You cannot 'run' on the programmer or program using the Reader. The bit-by-bit programming principles are unchanged, although programming is carried out by using large push switches. This means programs can be entered much more rapidly than when using the small DIP switch on the standard controller. Maximum program length is 99 lines. As the program is stored on the SmartCard it is retained when the battery is removed.

#### **Outputs**

The system supports six digital outputs. The outputs are numbered from right to left as opposed from left to right. An ULN2003 Darlington driver chip (included) on the Controller provides the 'open collector outputs', and hence replaces the need for the BCX38B output transistors. The back EMF protection diodes are also built into the chip. There is no pcb space for relays, although they may be attached by flying leads if desired. The controller includes a noise suppression choke which reduces the noise problems that can be created when using small DC motors. However DC motors should still have the noise suppression capacitors attached to their contacts.

#### **Inputs**

The system supports two pause inputs, and one reset input. The condition of the pause inputs is shown by yellow LED 'flags' on the Reader module. Each input can be directly activated by using the switches present on the Reader module, or by attaching your own switches on flying leads.

#### **Run Speeds**

The system supports six run speeds (0.1, 0.5, 1, 2, 5, 10s). With this system the run speed is programmed into the SmartCard memory using the Programmer module, and so no DIP switches are used to select the speed on the Reader. There is also no 'run' switch on the Reader module - the program starts running as soon as the SmartCard is inserted. The controller uses a high resolution ceramic oscillator and so there is no need for a 'speed adjust' preset resistor.

### **Power supply**

The modules can use power supplies from 6 to 12V. The supply voltage is the voltage that will appear on the open collector outputs (the digital outputs will be at 5V) of the Reader module. Please note we recommend the use of 6xAA cells (9V, or 7.2V with rechargeable cells) as opposed to 4xAA (6V) batteries as used on the standard controller. A 7.2V rechargeable ni-cad 'racing pack' as used in radio-controlled cars is ideal.

### **Computer Programming**

The 'Bit-by-Bit' SmartCard system can also be programmed via the 'Bit-by-Bit' (v2.0 or later) or Smart Link (v1.0 or later) computer software.

*For further details see the 'Bit-by-Bit Programming' section.*

## SMART LINK SOFTWARE INSTALLATION

These instructions are for installation on PCs that run the 'Windows' operating system. Details on how to install the software for other computer platforms will be provided with the appropriate installation disk.

Due to our constant development program, the software supplied on the installation disk may not be the latest version. Details of any changes, together with the latest version for downloading, are available from the TEP World Wide Web pages on the Internet.

### Minimum Requirements:

- Windows 3.0 or later
- 1.2 MB available hard-disk space
- 4 MB RAM
- Floppy disk drive.

### **Windows 3.x**

1. Place the floppy disk in the disk-drive.
2. From the 'Program Manager' choose 'File' menu and select 'Run'
3. Type **a:\setup** (enter)
4. Follow the on-screen instructions.

### **Windows 95**

1. Place the floppy disk in the disk-drive.
2. From the 'Taskbar' click on 'Start' and then select 'Run'
3. Type **a:\setup** (enter)
4. Follow the on-screen instructions.

TECHNICAL INFORMATION

This information may be required when installing the software on a network such as RM LM.

The following files should be installed in the \smrtcard directory:

- smrtcard.exe
- smrtcard.hlp

The following file should be installed in the \windows directory:

- smrtcard.ini

The following files should be installed in the \windows\system directory:

- vbrun300.dll
- cmdialog.dll
- cmdialog.vbx
- design.vbx
- grid.vbx
- msstat.vbx

*(These files are standard Visual Basic 'runtime' files, some of which may already be installed on your system - if so there is no need to update them).*