Repurposing an HP Calculator
Lab 2: Listening to the Keyboard
Computer Science and Computer Engineering Gateway Project

Stephen A. Edwards

Fall 2011

Abstract

In this lab, you will write software that will read the keyboard on the HP 20b and display which key is pressed.

The Matrix Keyboard

Figure 3 shows the schematic for the HP 20b's keyboard, which is a standard matrix type consisting of row and column wires that can be shorted together by the keys. It is connected to pins on the SAM7L chip that can be driven by a parallel I/O controller: a peripheral that enables software to control and read the state of each pin. Figure 1 is a more concise table that lists the pair of pins that each key shorts together when pressed.

The *lab2.tar.gz* file (on the course website) adds *keyboard.h* and *keyboard.c* to the files from *lab1.tar.gz*. Figure 2 lists the keyboard-related functions defined in *keyboard.c*.

To set up the peripherals to read the keyboard, call *keyboard_init*. This turns pins PC0–PC6 (the "columns") into outputs and pins PC11–PC15, and PC26 into inputs with pull-up resistors. The pull-up resistors ensure that if no key along a row is shorted the pin will have the value 1.

To see if a particular key is pressed, set the key's column to 0, every other column to 1, and read the key's row. If the key is pressed, the key's row will return 0, otherwise it will return 1. The easiest way to do this is to call the helper functions in *keyboard.h*.

main.c contains a program that reads the topmost keys on the keyboard and reports when they are pressed.

What To Do

- Write a function *keyboard_key* that returns a code that indicates either which key is currently being pressed or that no key is being pressed. Add this function to *keyboard.c*.
- Modify the code in *main.c* to use this function to report which key is being pressed.

"columns"	"rows"	PC11	PC12	PC13	PC14	PC15	PC26
	PC0	N	I/YR	PV	PMT	FG	Amort
	PC1	CshFl	IRR	NPV	Bond	%	RCL
	PC2	INPUT	(	)	+/−	←	
	PC3	▲	7	8	9	÷	
	PC4	▼	4	5	6	×	
	PC5	shift	1	2	3	−	
	PC6		0	.	=	+	

Figure 1: The HP 20b's keyboard layout. When pressed, each key shorts two pins: one for its column, one for its row.

```
// Initialize the keyboard and set all columns high
// with pullups on the rows
extern void keyboard_init(void);

// Set the given column high
extern void keyboard_column_high(int column);

// Set the given column low
extern void keyboard_column_low(int column);

// Return true if the row is high, false otherwise
extern int keyboard_row_read(int row);
```

Figure 2: Keyboard helper function declarations from *keyboard.h*

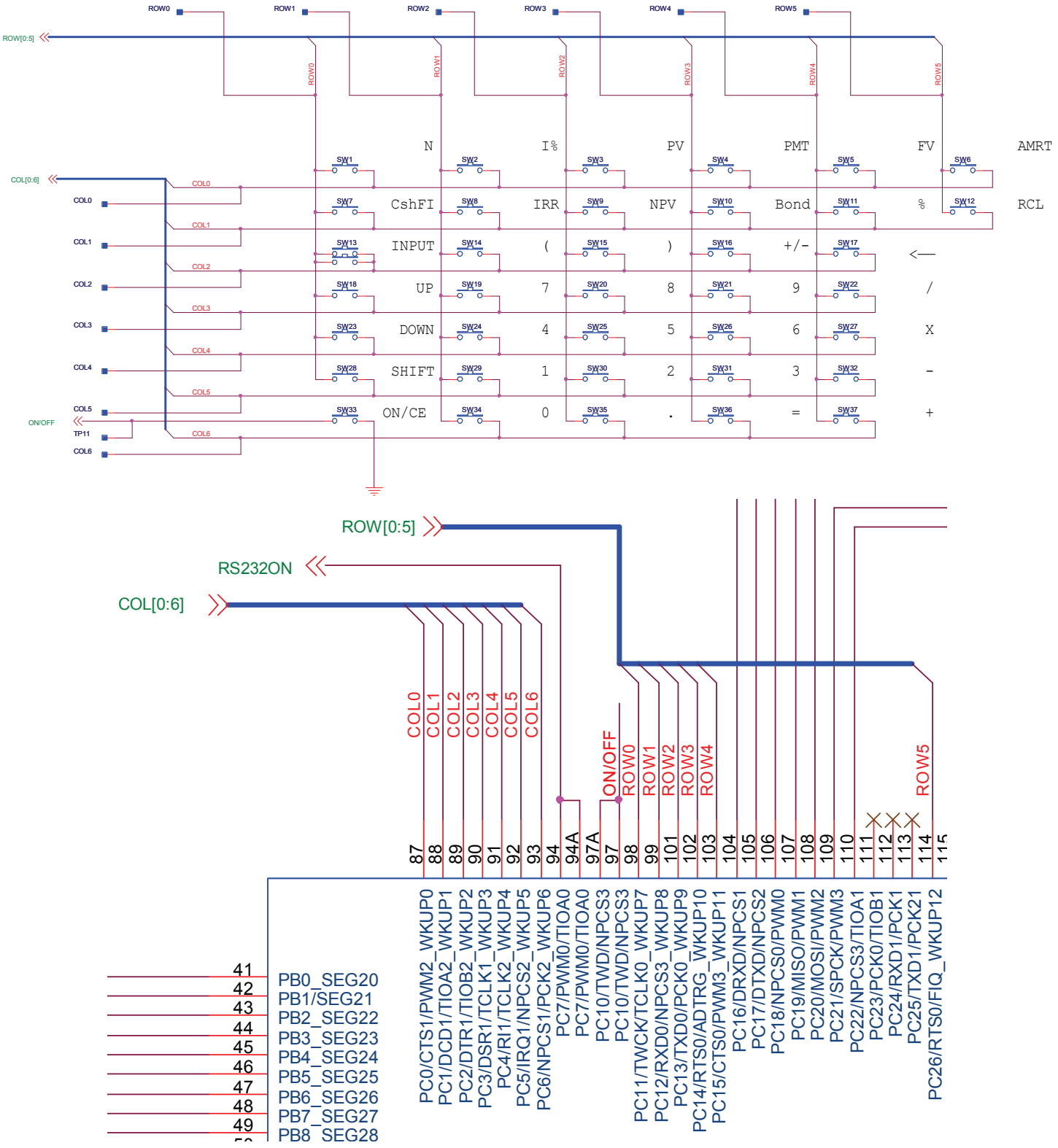

Figure 3: Schematics for the HP 20b keyboard. The top is the keyboard matrix itself; the bottom shows how the matrix is connected to the SAM7L chip. Note that the ON/CE key is separate (not part of the matrix) and that "columns" of keys are actually horizontal.