Making a Personal Digital Assistant on a Gameboy Advance

Problem

Powerful Personal Digital Assistants (PDAs) and smart-phones are becoming common place and are increasingly used to entertain their users with games. This is fine for those that predominately want an organiser and think of games as a bonus, but what about those that prefer the games, but would still like some organisational capability? Portable games consoles are powerful enough to run simple PDA-like applications but they lack the connectivity that PDAs and smart-phones have in abundance.

Objectives

To create an interface between a PC and a Nintendo Gameboy Advance (GBA) system which is as easy to use as any existing PDA and to create some proof-of-concept software that shows the use of this interface to communicate between the two.

Breakdown

- A GBA<->PC interface which requires no special drivers to be installed on the PC or GBA, and which in the presence of suitable software, communicates effectively between the two devices.
- A proof-of-concept application pair which demonstrate the effectiveness of the solution.

If time allows:

- The applications will be expanded to work with real data, such as the to-do lists / contact list of larger applications such as MS Outlook using SyncML.

Methods

Pre-requisites phase

Reading up on the subject and evaluating options before settling on a path.

- Learning how to effectively program the multi-player port on the GBA hardware and choosing the more appropriate mode to use.
- Choosing and evaluating the PC port to use for interfacing. Currently favoured is USB for its wide acceptance and easy-to-install end-user products.
- Choosing a microcontroller to interface between the two devices, which will be used to translate signals encapsulated in a standard-compliant protocol from the PC to a (simple) protocol of my own devising on the GBA.

Maturing phase

Becoming acquainted with the hardware:

- Probing the GBA hardware and testing multi-player port controlling code while determining the properties of the signals (voltage etc)
- Probing the chosen PC interface and testing code which manipulates this while determining the properties of the signals (voltage etc)

Once the results from the above are predictable and well understood:

- Create some signal translation circuits as applicable (perhaps the voltages are different for each device, the would be taken care of here)
- Probing / coding the microcontroller with some test data and getting a feel for the capabilities of the device (response times, speed, caveats)

Implementation phase

Bring the components together:

- Assemble the interface made up of the GBA connector cable, PC connector cable and microcontroller.
- Setup microcontroller to be identified and used by the PC (as applicable to the chosen PC port – i.e. USB will need PnP information, protocol setup etc).
- NB that the GBA does not require and special set-up as it does not have an operating system. Instead it is only required that the multi-player port chipset is in the correct mode (chosen during pre-requisites phase)

Create some proof-of-concept software which makes use of the new connectivity of the GBA such as synchronising a to-do list, contact list or similar task.

- Code an application for the GBA which displays incoming data on-screen.
- Code an application for the PC which generates test data and sends it to the GBA.

Once this is working:

- Code applications to do the reverse (PC -> GBA)
- Combine these into single bi-directional applications which swap test-data.

If time allows:

- The protocol of the GBA will be improved to resemble a more orthodox protocol.

Timetable

Checkpoint	*Description*
End of wk 5	Chosen PC port
End of wk6	Finished reading up on GBA multiplayer port, experimentation/probing started
	Chosen microcontroller
End of wk7	Experiments / probing PC port underway
End of wk8	Start designing/profile needs of translation circuits
End of wk9	Microcontroller experiments underway
End of wk2 Term 2	Interface able to (partially) transmit / receive data
End of wk4 Term 2	Interface stable and working
	Start development of applications
End of wk6 Term 2	Minimal applications complete

	Test on other PCs (non-development ones) for installation issues and compatibility
Mid Easter break	Project report is in draft essay form which contains all relevant information, but may been reworking.
Thurs wk 2, term 3	Submit Project report.

Resources

The following will be required in addition to those currently available to me within the department (such as the Linux workstations):

- Direct access to the parallel port on at least one workstation (for programming the GBA hardware using my personal 3rd-party programmers)
- Direct access to the serial port on at least one workstation (for debugging the GBA hardware in real-time using my personal 3rd-party debugger)
- Frequent access to a PIC-programmer (for the interface microcontroller)
- Manuals (possibly online) which explain how to use the PIC-programmer
- Manuals / reference on programming PICs (dependant on the brand of PIC chosen, it is expected that the project supervisor will guide this choice to best suit the resources available)