FH Science Day 2008
ISBN 978-3-8322-7643-0

Packet Sniffer for the Physical Layer of the Single Wire Protocol

Michael Roland, Christian Saminger, Josef Langer

Upper Austria University of Applied Sciences, School of Informatics, Communications and Media, Softwarepark 11, 4232 Hagenberg, AUSTRIA

ABSTRACT

The Single Wire Protocol (SWP, ETSI TS 102 613) is intended as direct interface between a mobile phone's SIM card (UICC) and the mobile phone's contactless front-end (CLF). The SWP's final technical specification has just been released. The first devices implementing this communication protocol, mainly in its draft versions, are already in production. As a consequence there will be a demand for a test suite implementing a reference design and test methods for both the SWP master and the SWP slave.

With communication protocols it is usually important to debug communication problems between multiple devices. One way to trace and decode the transferred data packets are packet sniffers. These systems contain hardware components and software implementations to wiretap and analyze the physical interface of the connection, to capture the data and to decode the packets into human readable information.

The SWP uses a single wire for full-duplex communication between one master and one slave device. While master-to-slave data transfers take place in the voltage domain, slave-to-master data transfers take place in the current domain. In a first step, this paper discusses approaches to intercept the communication on the SWP's data wire without influencing the actual communication.

The information tapped from the SWP's data wire is still difficult to be read by hand. Thus, in a second step, a method for retrieving the state of the single wire interface is developed. Moreover, this paper gives an overview on how to decode the data link layer communication from the intercepted data streams.

Contact: michael.roland@fh-hagenberg.at, christian.saminger@fh-hagenberg.at, josef.langer@fh-hagenberg.at

1 INTRODUCTION

A steadily increasing number of mobile phones are equipped with Near Field Communication (NFC) technology. With this progress a central question arises: Who should have control over the secure element? With current applications the secure element is an integrated part of the cellular phone. This leaves control to the phone manufacturers [1]. For telephony, the counterpart to the secure element is the subscriber identity module (SIM). The SIM is part of the universal integrated circuit card (UICC). This card is already present in the cellular phone. Therefore, it would be reasonable to also use the UICC as the secure element for NFC and, thus, shift the control to the mobile network operators (MNOs) or a newly established trusted service manager (TSM).

To implement the UICC as the secure element for NFC, a direct interface to the mobile phone's contactless front-end (CLF) is necessary. Major vendors have agreed that the newly developed Single

ISBN 978-3-8322-7643-0

Wire Protocol (SWP) should be used for this communication. The SWP's final technical specification [2] has just been released. First devices like UICCs, NFC front-end integrated circuits and NFC-enabled microcontrollers implementing the SWP standard are already announced and in production.

Testing these devices requires test suites such as reference designs, test methods and also a means of intercepting the SWP data transmission. This paper describes methods of wiretapping the SWP interface and evaluating that data.

2 SINGLE WIRE PROTOCOL

The SWP is a full-duplex serial communication protocol. It is intended for direct data transfer between a CLF and a UICC. A UICC has eight contacts. Only one of these contacts is unused with previous specifications [3]. As a result, the SWP is required to use only this contact (SWIO) to be backward compatible to existing standards and applications.

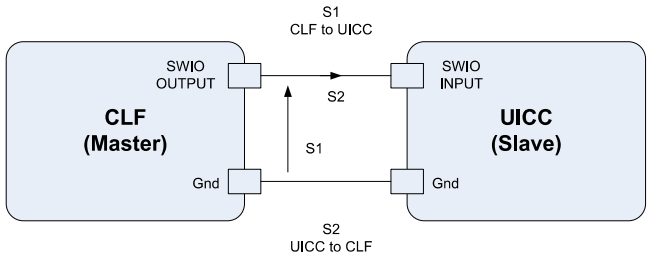

Figure 1. Single Wire Protocol: Data is transferred through a single wire (with a reference to ground), where S1 is the master-to-slave signal in the voltage domain and S2 is the slave-to-master signal in the current domain [2].

Figure 1 shows the principle of the SWP data transmission: While master-to-slave data (S1) is transmitted in the voltage domain, slave-to-master data (S2) is transmitted in the current domain on the same wire. Combined with a special modulation scheme for S1, this enables full-duplex communication between the two devices.

The SWP is divided into three layers:

(1) The physical transmission layer (PHY) specifies the bit coding and the SWP interface's state management.

(2) The medium access control layer (MAC) defines the bit order, the framing, the bit stuffing and an error detection mechanism.

(3) The logical link control layer (LLC) defines the format of data packets and several protocols for data exchange.

2.1 Master-to-Slave Signaling (S1)

Master-to-slave communication uses a pulse width modulation bit coding as shown in Figure 2. A logical one is encoded by keeping the voltage signal on SWIO high for 75 percent of the bit-duration and low for the rest of the bit-duration. A logical zero is encoded by keeping the voltage signal on SWIO high for 25 percent of the bit-duration and low for the rest of the bit-duration. The bit-duration may vary for each transmitted bit. There are three different states for the signal S1:

ISBN 978-3-8322-7643-0

- The voltage signal is constantly low when the SWP interface is disabled.
- The voltage signal is constantly high when the SWP interface is enabled without data being transmitted.
- The voltage signal is modulated when the SWP interface is active with data being transmitted. The signal S1 is also used as bit clock for slave-to-master signaling.

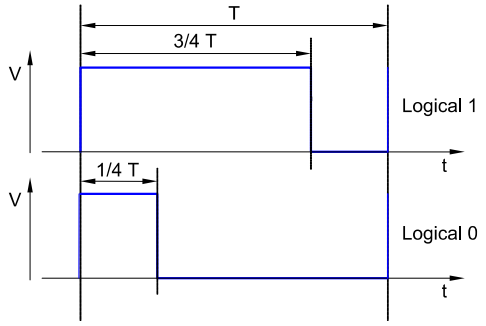

Figure 2. Pulse width modulation bit coding is used for master-to-slave (S1) communication [2].

2.2 Slave-to-Master Signaling (S2)

Slave-to-master communication uses NRZ-L (non-return-to-zero-level) bit coding. Additionally, as a result of transmitting data in the current domain on the same wire, the slave can only send data during the high pulses of the voltage signal. A logical one is encoded by sinking a current I_H on then SWIO contact. A logical zero is encoded by sinking a current I_L on the SWIO contact. Table 1 shows the allowed current ranges as specified by [2].

Table 1. Allowed ranges for current signal S2 according to [2].

Symbol	State	Minimum	Maximum
I_H	Logical 1	600 µA	1000 µA
I_L	Logical 0	0 µA	20 µA

There are two different states for the signal S2:

- The current signal is constantly low when no data being transmitted on S2.
- The current signal is modulated when data being transmitted on S2.

2.3 States of the SWP interface

The SWP has three different states:

- In *deactivated* state the SWP interface is disabled and no communication takes place on SWIO. Only the master is able to enable the SWP interface.
- In *suspended* state the SWP interface is enabled but no communication takes place on SWIO. Both, the master and the slave, may activate the SWP interface to transmit data.
- In *activated* state the SWP interface is enabled and communication takes place on SWIO.

ISBN 978-3-8322-7643-0

2.4 Medium Access Control

The SWP is a block-oriented protocol. As shown in Figure 3, each frame begins with a start-of-frame (SOF) flag, followed by the payload and its checksum, and ends with an end-of-frame (EOF) flag. Bit stuffing is used to distinguish the data from the SOF and EOF flags. Therefore, a logical zero bit is inserted after five consecutive logical ones.

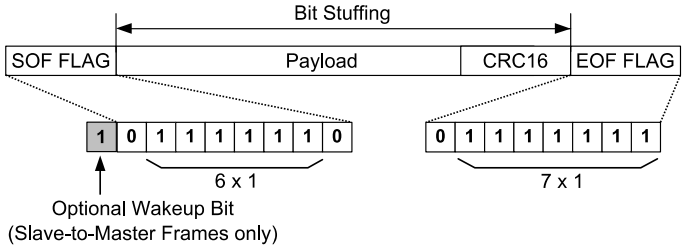

Figure 3. Frame structure used for data exchange on the MAC layer between master and slave [2].

2.5 Logical Link Control

An SWP frame's payload is called link protocol data unit (LPDU). The logical link control layer defines the format of the LPDU. Moreover, it specifies measures for error correction and flow control. Three different LLC protocols, the ACT protocol, the CLT protocol and the SHDLC protocol, exist. Decoding the LLC data is beyond the scope of this paper.

3 WIRETAPPING AND PACKET SNIFFING

Wiretapping is a term commonly referring to the monitoring of telephone calls. Just as with telephone lines, wiretapping can be used to monitor virtually any wired data link. Packet sniffing is a method of eavesdropping every frame of data that is transferred between two or more endpoints [4]. A bus analyzer combines both techniques. Therefore, it intercepts the electrical signal on the data wire, analyzes the state of the communication and receives the data frames. Furthermore, a typical bus analyzer contains a protocol analyzer which decodes each frame's payload into LLC packets or even into application layer data.

Bus analyzers and packet sniffers are usually used to debug communication problems. They can be employed for detecting errors concerning the framing, the bit stuffing, the checksum generation or the SWP interface state. Moreover, they can be used to identify problems with LLC layer or higher layer communication. Another useful side-effect of packet sniffing is that the whole data traffic can be logged.

ISBN 978-3-8322-7643-0

4 SWP PACKET SNIFFING SYSTEM

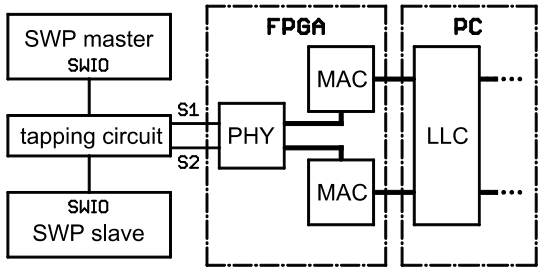

Figure 4. The SWP packet sniffing system consists of a tapping circuit, an FPGA with the physical transmission layer and the medium access control layer, and PC software for decoding higher layers.

A hardware/software co-design approach is used for the SWP packet sniffing system. The system architecture is shown in Figure 4. The tapping circuit is designed with discrete ICs. The physical transmission layer and the medium access control layer are implemented on a field-programmable gate array (FPGA). The higher layers' packets are decoded with software on the PC.

4.1 Tapping the Physical Layer

There are several approaches for intercepting the signal on the SWIO contact. The following sections describe two methods which have been considered especially useful.

4.1.1 *Measuring Current and Voltage on SWIO*

There are two signals on the SWIO wire, the voltage signal S1 and the current signal S2. The physical transmission layer is implemented on an FPGA. Thus, the analog signals have to be converted into digital signals. The conversion circuit is shown in Figure 5.

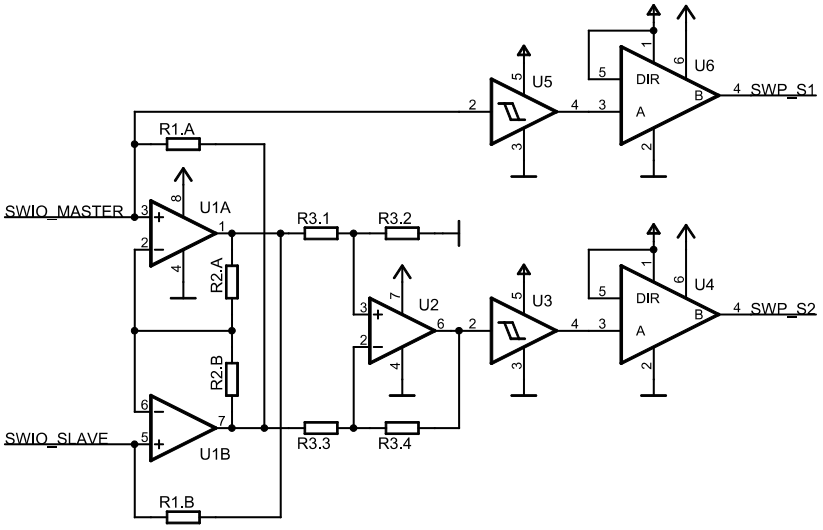

Figure 5. The measuring probe is composed of an ampere meter with a low voltage drop [5], [6] and Schmitt triggers to form the analog voltage signals into digital signals. Bus transceivers are used to shift the voltage levels of the digital signals to those of the FPGA. ($R_{1.A} = R_{1.B} = R_1$, $R_{2.A} = R_{2.B} = R_2$, $R_{3.1} = R_{3.2} = R_{3.3} = R_{3.4} = R_3$)

ISBN 978-3-8322-7643-0

The voltage signal S1 is formed into a binary digital signal. The SWP specification [2] states that signal levels on SWIO must comply with those of standard 1.80-volt-logic components except that voltages up to 3.3 volts must be tolerated. Therefore, a Schmitt trigger from 1.80 volt logic series that is 5-volt-tolerant is used. The output voltage level is then shifted to be compatible to the inputs of the FPGA.

To form the current signal S2 into a binary digital signal, in a first step, the current I_S on the SWIO wire is transformed into a voltage signal. Measuring the current must not influence the signaling. As a result, an ampere meter with a low voltage drop [5], [6] is used to perform this task. Three operational amplifiers are used to measure the current between the clamps SWIO_MASTER and SWIO_SLAVE. The voltage drop between these two clamps is approximately zero and the current sunk on SWIO_MASTER is virtually the same as the current sourced on SWIO_SLAVE. The output voltage V_A of the subtracting amplifier (U2) is

$$V_A = \left(R_{1.A} + R_{1.B}\right) \cdot I_S = 2 \cdot R_1 \cdot I_S . \tag{1}$$

In a second step, the voltage signal V_A is converted into a binary digital signal in the same way as it is done with signal S1. The resistors $R_{1.A}$ and $R_{1.B}$ are adjusted to adapt the range of V_A to the input range of the Schmitt trigger.

4.1.2 *Using an Analog SWP Front-end*

Another way to intercept the signal on SWIO is the use of an analog SWP front-end (Figure 6).

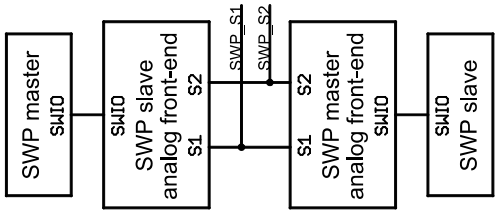

Figure 6. The packet sniffing circuit is composed of a master's and a slave's analog front-end.

The SWIO contact of the SWP master is connected to a slave's analog front-end and the SWIO contact of the SWP slave is connected to a master's analog front-end. The digital sides of the SWP front-ends are linked-up to each other. These digital signals can then be used on the FPGA.

With a slight modification of this circuit, the signals can be fed through the FPGA. This approach allows the signals to be modified on the FPGA. Therefore, in addition to packet sniffing, packets could also be modified. Moreover, new packets could be induced into the system. Consequently, an SWP master or an SWP slave could be emulated.

The major disadvantage with this approach is that the electrical characteristics and the timing of the signals are altered. The slave does not receive the original master's signal and vice versa. Instead, that signal is generated by the intermediate analog front-end. As a result, if the intermediate analog front-ends are more tolerant about the electrical characteristics of the signal, problems with the signal levels might disappear while the tapping circuit is connected. Moreover, if there is an additional signal delay, the signal S2 will no longer be synchronous to S1. Thus, the signal delay leads to invalid values of S2. Although the master might still be able to properly receive S2 if it is delayed for a short time, an additional delay violates the requirements of the SWP specification [2].

ISBN 978-3-8322-7643-0

4.2 Recovering the State of the SWP Interface

The physical transmission layer and the medium access control layer are implemented on an FPGA. In a first iteration, the digital signals S1 and S2 are decoded into bit streams. A finite state machine is used to track the state of the SWP interface. The state transitions are determined by evaluating S1. Thus, when S1 is constantly low the interface is considered to be *deactivated*. When S1 is constantly high the interface is considered to be *suspended*. And when a bit clock is transmitted on S1 the interface is considered to be *activated*. During activated state bits on S1 and S2 are decoded and passed on to the MAC layer. For the signal S1, the durations of the high and low phases are compared to calculate logical ones and zeros according to Figure 2. For the signal S2, bits are sampled during the high phases of S1.

In a second step, medium access control layer frames are extracted from the bit streams. For this purpose the streams are scanned for the start-of-frame (SOF) and end-of-frame (EOF) flags. For the intermediate data the bit stuffing procedure is reversed. The frame is then decomposed into its payload and checksum. This information is passed on to the PC for higher layer protocol decoding.

The packet sniffer has to be tolerant towards errors. Consequently, one of the components which have to be fault-tolerant is the bit de-stuffing procedure. As the SOF and EOF sequences are intended bit stuffing errors, unintended errors lead to framing errors. The resulting fragments have to be passed up to packet logger without higher layer processing. Another component that has to be fault-tolerant is the payload and checksum extraction. Usually frames with an invalid checksum are supposed to be dropped. For the packet sniffer these frames are of special interest as they are usually a source of packet loss. Therefore, packets have to be passed on to the PC for higher layer protocol decoding regardless of their checksum.

4.3 Transmitting LLC Layer Packets to the PC

The extracted LPDUs, their checksums and some status information about the interface, like the state of the SWP interface and detected errors, must be transferred to the PC. On the PC this data can be decoded into higher layer information and can be displayed to the user. For transmitting this data an appropriate interface is necessary.

The minimum bit-duration of the SWP is 590 ns. Therefore, the maximum throughput per signal is about 1.7 megabit per second (Mbps). Although the full-duplex capabilities of the SWP are not used in normal operation, the packet sniffer must be able to cope with unexpected events on the interface. And thus, it must be able to receive full-duplex communication. As a consequence, the interface between the FPGA and the PC requires a bandwidth of about 4 Mbps to transmit data and status information.

5 SUMMARY

This paper presents a system for sniffing the physical layer of the Single Wire Protocol. It illustrates the various aspects that have to be considered during the design and implementation of the packet sniffer.

Two methods for tapping the physical interface were evaluated and their advantages and disadvantages were shown. A prototype for the promising circuit for measuring current and voltage on SWIO was implemented and its proper operation could be verified.

In the next step, a procedure for decoding the signals by deriving the state and the bit clock from S1 is developed. Then, the measures for extracting MAC frames from the data streams are explained. Attention was focused on the fault-tolerance of the decoding procedure.

Finally, the requirements for the interface between the FPGA and the PC are defined. For a bandwidth of about 4 Mbps some high speed interface like USB is necessary.

ISBN 978-3-8322-7643-0

REFERENCES

[1] B. Ray, “The future of the SIM hangs by a single wire,” The Register, February 2008.

[2] Smart Cards; UICC - Contactless Front-end (CLF) interface; Part 1: Physical and data link layer characteristics (Release 7), European Telecommunications Standards Institute Std. ETSI TS 102 613, Rev. 7.1.0, February 2008.

[3] Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 7), European Telecommunications Standards Institute Std. ETSI TS 102 221, Rev. 7.10.0, February 2008.

[4] S. Ansari, S. G. Rajeev, and H. S. Chandrashekar, “Packet sniffing: A brief introduction,” IEEE Potentials, vol. 21, issue 5, pp. 17–19, December 2002/January 2003.

[5] U. Tietze, and C. Schenk, Halbleiter-Schaltungstechnik, 11th ed., Springer Verlag Berlin Heidelberg New York, 1999.

[6] R. Lerch, Elektrische Meßtechnik, Springer Verlag Berlin Heidelberg, 1996.