Programming manual for the TIRIS library for the PHL-2700

Introduction

This is a preliminary manual giving information about the functions in tiris.lib that give access to the TIRIS read capabilities of the PHL-2700 with a TIRIS RFID-module that makes it possible to read 134 kHz TIRIS transponders

How to build an application using the library

To build an application, see the programming manual for the PHL-2700. The functions of the RFID library can be linked by adding the following line to the link file, e.g. in front of the line “LOAD c:\mccm77\CM77ISLC.LIB”

```
LOAD TIRIS.LIB
```

In all the source files that make use of the functions of the RF library, include the file “tiris.h”.

Note: use operating system CBW99990. This operating system implements a special baudrate of 16.000bps for the reading of the TIRIS transponders

The TIRIS library supports 4 main types of TIRIS transponders:

Read-only transponders (RO)	(RI-TRP-Rxxx)
Read-write transponders (RW)	(RI-TRP-Wxxx)
Multipage transponders (MPT)	(RI-TRP-Cxxx)
Selective Addressed Multipage Transponders (SAMPT)	(RI-TRP-Ixxx)

The functions in the TIRIS library are described in the following pages.

In the appendixes general information about the TIRIS transponders is given.

General functions

tiris_init

Description switches the TIRIS RFID-module on, establishes communication by setting the correct baudrate and requesting (and checking) the diagnostic byte from the RF Module IC (TMS3705A) .

Syntax
```
int tiris_init(void);
```

Arguments -

Returns

- on success	OK
- on no diagnostic byte received	NO_RESPONSE
- on wrong diagnostic byte received	HARDWARE_FAILURE
- on error opening COM4	UNKNOWN_ERROR

Remarks
- if the TIRIS device is already on, the function does nothing and returns OK.
- a return value of UNKNOWN_ERROR means that COM4 cannot be opened. Upgrade the OS to a version that does support COM4.
- to indicate that the TIRIS RFID module is active it is suggested to turn the green or red LED on.
- do not leave the TIRIS RFID module on longer than necessary, because it increases power consumption.

This function must always be executed before calling any of the read, program or lock functions below, otherwise they won't work.

tiris_close

Description switches the TIRIS RFID module off and closes the comport

Syntax
```
int rfidclose(void)
```

Arguments -

Returns

- on succes	OK
- on error closing COM4	UNKNOWN_ERROR

Remarks
- call this function whenever done reading an TIRIS transponder, to reduce power consumption.
- if the error value UNKNOWN_ERROR is returned, make sure that the OS supports COM4. If not, upgrade the OS to a version that does support COM4
- iIf the TIRIS RFID-module was already closed, the function does nothing.
- if the green or red is used to indicate when the TIRIS RFID-module is active, turn it off on closing the TIRIS RFID-module.

This function must always be executed whenever done using the read, program or lock functions below. Otherwise they TIRIS RFID-module will be left on and keeps consuming power

tiris_lib_version

Description returns the library version string of this TIRIS library

Syntax
```
char *tiris_lib_version(void)
```

Arguments -

Returns
- pointer to a buffer, into which the version string is put

Remarks
- the buffer size is 8 characters.
- the string is not zero-terminated.

tiris_antenna_power

Description Sets the antenna power of the TIRIS RFID-module on or off

Syntax `void tiris_antenna_power(unsigned int mode)`

Arguments

`mode`	0: antenna power off 1: antenna power on

Returns -

Remarks

- this function is intended for diagnostics and adjustment purposes.
- execute the function ‘tiris_init’, to turn on the TIRIS RFID-module, before executing this function. Afterwards use the function ‘tiris_close’ to turn off the module.

Read functions

tiris_charge_only

Description Gets the 64 bits of identification data of page 1 of a TIRIS transponder within range and places it in the global buffer: recv_data[].

Syntax `int tiris_charge_only(void)`

Global `extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments -

Returns

- on read success of read-only transponder	READ_ONLY_OK
- on read success of read-write transponder	READ_WRITE_OK
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range)	NO_RESPONSE
- on general error or unexpected response	UNKNOWN_ERROR

Remarks

- this function is supported by the following 3 types of transponders:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
  - *Multipage transponders (MPT)*
- the function is called tiris_charge_only, because no explicit data is send to the transponders. The transponders are only charged and discharged. After being charged the transponders send there identification data of page 1 back during the discharge period.

tiris_read_page

Description Reads the requested page of data from a TIRIS transponder. On success the 64 bits of identifaction data of that page is places in the global buffer: recv_data[]

Syntax `int tiris_read_page(int page_nr)`

Global `extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments `page_nr` number of the page that must be read

Returns

- on success, requested page read and page wasn't unlocked	UNLOCKED_PAGE_READ_OK
- on success, requested page read and page was locked	LOCKED_PAGE_READ_OK
- page 1 read of read-only transponder or requested page of multipage transponder was read and was read-only	READ_ONLY_OK
- page 1 read of read-write transponder	READ_WRITE_OK
- no identification data in requested page	NO_ID_DATA_IN_PAGE
- the returned (un)locked page wasn't the requested page	(UN)LOCKED_PAGE_ADDRESS_ERROR
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range)	NO_RESPONSE
- on general error or unexpected response	UNKNOWN_ERROR

Remarks

- This function is supported by the following 2 types of transponders:
  - *Multipage transponders (MPT)*
  - *Selective Programmable Addressable Multipage transponders (SAMPT)*

- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*

- Note that page 1 of a multipage transponder contains an unique identifier that's factory programmed and is a read-only page.

tiris_selective_read_page

Description Reads the requested page of data of the selectively addressed transponder. On success the 64 bits of identifaction data of that page is places in the global buffer: recv_data[].

Syntax
```
int tiris_selective_read_page(int page_nr,
                              unsigned char *sel_addr)
```

Global
`extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments

`page_nr`	number of the page that must be read
`sel_addr`	pointer to 3 bytes of data that contain the selective address of the transponder that must be read (sel_addr[0] = LSB)

Returns

- see return values of tiris_read_page	see tiris_read_page
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range or selective address didn't match)	NO_RESPONSE

Remarks

- This function is supported by the following type of transponders:
  - *Selective Programmable Addressable Multipage transponders (SAMPT)*
- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
- The selective address of the SAMPT can be received by using the function tiris_read_page(page_nr) with page_nr = 1. After this function was successfully executed, the selective address will be the first three bytes of data in the receive buffer: recv_data[] (recv_data[0] = LSB of selective address)

Program functions

tiris_program_page

Description Programs the requested page of a TIRIS transponder with the given data. On success the 64 bits of the new identifaction data of that page are places in the global buffer: recv_data[].

Syntax `int tiris_program_page(int page_nr, unsigned char *data)`

Global `extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments

`page_nr`	number of the page that must be programmed
`data`	pointer to 8 bytes of data that must be programmed in the requested page

Returns

- on success, requested page was successfully programmed	PAGE_PROGRAMMED_OK
- page 1 was read of a read-only transponder or the requested page of a multipage transponder was read-only	READ_ONLY_OK
- page 1 was read of a read-write transponder	READ_WRITE_OK
- on no success, requested page was already locked	PAGE_WAS_LOCKED
- requested page was not locked due to insufficient field strength	INSUFFICIENT_FIELD_STRENGTH
- result of the programming operation is not reliable, verify the data to be sure the programming was successful	RESULT_NOT_RELIABLE
- no identification data in requested page	NO_ID_DATA_IN_PAGE
- returned (un)locked page wasn't requested	(UN)LOCKED_PAGE_ADDRESS_ERROR
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range)	NO_RESPONSE
- on general error or unexpected response	UNKNOWN_ERROR

Remarks

- This function is supported by the following type of transponders:
  - *Multipage transponders (MPT)*
- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
- Note that page 1 of a multipage transponder contains its unique identifier that's factory programmed and is a read-only page.
- It is recommended that the result of the programming operation is always verified by checking the returned data in the receive buffer: recv_data[]. If the returned data isn't correct, the programming operation should be repeated untill the result is correct.

tiris_selective_program_page

Description Programs the given page of the selectively addressed transponder with the given data. On success the 64 bits of the new identifaction data of that page is returned in the global buffer: recv_data[]

Syntax
```
int tiris_selective_program_page(int page_nr,
                                 unsigned char *sel_addr,
                                 unsigned char *data)
```

Global

`extern unsigned char recv_data[13]`	the received data is places in this buffer

Arguments

`page_nr`	number of the page that must be programmed.
`sel_addr`	pointer to 3 bytes of data that contain the selective address of the transponder that must be programmed (sel_addr[0] = LSB).
`data`	pointer to 10 bytes of data that must be programmed in the requested page.

Returns

- see return values of tiris_program_page	see tiris_program_page
- on no response or invalid response from the transponder or the TIRIS RFID-module (i.e. tag wasn't in range or selective address was incorrect)	NO_RESPONSE

Remarks

- This function is supported by the following type of transponders:
  - *Selective Programmable Addressable Multipage transponders (SAMPT)*
- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
- The selective address of the SAMPT can be received by using the function tiris_read_page(page_nr) with page_nr = 1. After this function was successfully executed, the selective address will be the first three bytes of data in the receive buffer: recv_data[] (recv_data[0] = LSB of selective address)

tiris_program_RW

Description Programs a Read-Write transponder (RW) with the given data. On success the 64 bits of the new identifaction data is returned in the global buffer: recv_data[].

Syntax
```
int tiris_program_RW(unsigned char *data)
```

Global

`extern unsigned char recv_data[13]`	the received data is places in this buffer

Arguments

`data`	pointer to 10 bytes of data that must be programmed in the requested page.

Returns

- see return values of tiris_program_page	see tiris_program_page

Remarks

- This function is supported by the following type of transponders:
  - *Read-write transponders (RW)*
- This function causes the following type of transponders to return its identification data of page 1:
  - *Read-only transponders (RO)*

Lock functions

tiris_lock_page

Description Locks the given page of data of a transponder. On success the page is locked and cannot be unlocked anymore. The 64 bits of identifaction data of the locked page is places in the global buffer: recv_data[].

Syntax `int tiris_lock_page(int page_nr)`

Global `extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments `page_nr` number of the page that must be locked

Returns

- on success, requested page was successfully programmed	PAGE_LOCKED_OK
- page 1 read of read-only transponder or requested page to lock of MPT was read-only	READ_ONLY_OK
- page 1 read of read-write transponder	READ_WRITE_OK
- requested page was not locked due to insufficient field strength	INSUFFICIENT_FIELD_STRENGTH
- page probably hasn't been locked, verify the page to check if the page has been locked or not	PAGE_NOT_LOCKED
- no identification data in requested page	NO_ID_DATA_IN_PAGE
- returned (un)locked page wasn't requested	(UN)LOCKED_PAGE_ADDRESS_ERROR
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range)	NO_RESPONSE
- on general error or unexpected response	UNKNOWN_ERROR

Remarks

- This function is supported by the following type of transponders:
  - *Multipage transponders (MPT)*
- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
- If the requested page was already locked, this function will return PAGE_LOCKED_OK and not PAGE_WAS_LOCKED
- Note that page 1 of a multipage transponder contains its unique identifier that's factory programmed and is a read-only page, so it cannot be locked

tiris_selective_lock_page

Description Locks the given page of data of the selectively addressed transponder. On success the page is locked and cannot be unlocked anymore. The 64 bits of identifaction data of the locked page is places in the global buffer: recv_data[]

Syntax
```
int tiris_selective_lock_page(int page_nr,
                              unsigned char *sel_addr)
```

Global
`extern unsigned char recv_data[13]` the received data is places in this buffer

Arguments

`page_nr`	number of the page that must be locked
`sel_addr`	pointer to 3 bytes of data that contain the selective address of the transponder that must be locked (sel_addr[0] = LSB)

Returns

- see return values of tiris_lock_page	see tiris_lock_page
- on no response or invalid response from transponder or TIRIS RFID-module (i.e. tag not in range or selective address was incorrect)	NO_RESPONSE

Remarks

- This function is supported by the following type of transponders:
  - *Selective Programmable Addressable Multipage transponders (SAMPT*
- This function causes the following 2 types of transponders to return there identification data of page 1:
  - *Read-only transponders (RO)*
  - *Read-write transponders (RW)*
- If the requested page was already locked, this function will return PAGE_LOCKED_OK and not PAGE_WAS_LOCKED
- The selective address of the SAMPT can be received by using the function tiris_read_page(page_nr) with page_nr = 1. After this function was successfully executed, the selective address will be the first three bytes of data in the receive buffer: recv_data[] (recv_data[0] = LSB of selective address)

Appendix: Information about TIRIS Transponders

The following information is based on documentation provided by Texas Instruments.

Transponder versions

The TIRIS library supports 4 main types of TIRIS transponders:

- Read-only transponders (RO) (RI-TRP-Rxxx)
- Read-write transponders (RW) (RI-TRP-Wxxx)
- Multipage transponders (MPT) (RI-TRP-Cxxx)
- Selective Addressed Multipage Transponders (SAMPT) (RI-TRP-Ixxx)

The different types of transponders support different types of commands and have different memory organizations. In the table below can be seen which of 7 main functions of the TIRIS library can be used on which type transponder. It also shows which memory page(s) can be accessed. For a more detailed description about the use of the functions of the TIRIS library, see the chapters: 'Read, Program and Lock functions'

	RO	RW	MPT	SAMPT
Charge only ID response	Yes	Yes	Yes	No
Read Page	Page 1	Page 1	Page 1..17	Page 1..17
Selective Read Page	-	-	-	Page 1..17
Program Page	-	Page 1	Page 1..17	-
Selective Program Page	-	-	-	Page 1..17
Lock Page	-	-	Page 1..17	-
Selective Lock Page	-	-	-	Page 1..17

On the following pages, the memory organization of the 4 types of transponders is described in detail.

Memory organisation of Read-only transponders (RO)

The currently available Read-only transponders (RO) contain 1 page of data of 80 bits (10 bytes) in size.
64 bits of the 80 bits of data contain the Unique identifier of the transponder and the remaining 16 bits, called the Data Block Check Character (DBCC), are used as protection data. The protection data is the result of a CRC-CCITT protection algorithm that is used to protect and verify the vality of the identifation data.
This page of the transponder is read-only, so the unique identifier cannot be changed.
The unique identifier is factory-programmed and because it's 64 bits long, it has an address range of more than $1{,}8 * 10^{19}$ individual addresses. The possibility cannot be excluded that at some future time, a previously used address is assigned to a new transponder. Considering the expected lifetime of transponders however, and based on statistical calculations, the probability that 2 transponders with the same address are presented to the same identification system is practically nihil.

Memory organisation of the RO transponder

	LSB 80 bits	MSB
Page 1	64 bits Unique Identification data	16 bits data BCC

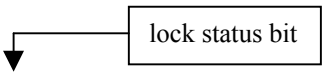
Memory organisation of Read-write transponders (RW)

The currently available Read-write transponders (RW) contain 1 page of data of 80 bits (10 bytes) in size.
64 bits of the 80 bits of data are used for identification data and the remaining 16 bits, called the Data Block Check Character (DBCC), are used as protection data. The protection data is the result of a CRC-CCITT protection algorithm that is used to protect and verify the vality of the identifation data.
The page of the transponder is readable and user programmable.

Memory organisation of the R/W transponder

	LSB 80 bits	MSB
Page 1	64 bits Identification data	16 bits data BCC

Memory organisation of Multipage transponders (MPT)
The currently available multipage transponders (MPT 0/17) contain 17 pages of data, each 80 bits (10 bytes) in size. Each page also has one lock bit, that determines the lock status of the pages.
64 bits of the 80 bits of data are used for identification data and the remaining 16 bits, called the Data Block Check Character (DBCC), are used as protection data. The protection data is the result of a CRC-CCITT protection algorithm that is used to protect and verify the vality of the identifation data.
Page 1 is read-only. The identification data of page 1 contains a unique address that is factory-programmed and is 64 bits long, which makes the probability that 2 transponders with the same address are presented to the same identification system practically nihil.
Pages 2 to 17 of the transponder are readable, user programmable and lockable. Note that once a page is locked it cannot be unlocked.



Memory organisation of the MPT 0/17

	LSB 80 bits	MSB	1
Page 1	64 bits Unique Identifier	16 bits data BCC	
Page 2	64 bits Identification data	16 bits data BCC	L
Page 3	64 bits Identification data	16 bits data BCC	L
Page 4	64 bits Identification data	16 bits data BCC	L
Page 5	64 bits Identification data	16 bits data BCC	L
Page 6	64 bits Identification data	16 bits data BCC	L
…	…	…	.
Page 17	64 bits Identification data	16 bits data BCC	L

Memory organisation of Selective Programmable Addressable Multipage transponders (SAMPT)
The currently available Selective Programmable Addressable Multipage transponders (SAMPT 0/17-24) contain 17 pages of data, each 80 bits (10 bytes) in size. Each page also has one lock bit, that determines the lock status of the pages.
64 bits of the 80 bits of data are used for identification data and the remaining 16 bits, called the Data Block Check Character (DBCC), are used as protection data. The protection data is the result of a CRC-CCITT protection algorithm that is used to protect and verify the vality of the identifation data.
The 64 identification bits of page 1 is divided into 24 selective address bits, which make up the selective address of the transponder and the remaining 40 bits are normal identification bits. The selective address can be used to read, program or lock a single specific transponder. This is useful in a environment in which more than one transponders can be presented to the reader at the same time.
Each page of the transponder is readable (selectively addressed and non-addressed), user programmable (selectively addressed) and lockable (selectively addressed). Note that once a page is locked it cannot be unlocked.

Memory organisation of the SAMPT 0/17-24

	LSB 80 bits		MSB	L
Page 1	24 bit selective adress	40 bit identification data	16 bits data BCC	1
Page 2	64 bits Identification data		16 bits data BCC	1
Page 3	64 bits Identification data		16 bits data BCC	1
Page 4	64 bits Identification data		16 bits data BCC	1
Page 5	64 bits Identification data		16 bits data BCC	1
Page 6	64 bits Identification data		16 bits data BCC	1
…	…		…	.
Page 17	64 bits Identification data		16 bits data BCC	1

Non-addressed and addressed operations

The reader module in the PHL-2700 supports both non-addressed and addressed operations.
It is key to successful implementation of the non-addressed operation that only one transponder is within the reader's range.
Otherwise, the following may occur:

- if the reader is performing a read function, 2 or more answers will be sent by transponders. These answers will collide, resulting in an unintelligible message at the reader's side.
- if the reader is performing a write function, all reachable transponders will perform this function, and for the reasons stated above, such that the reader will not receive a clear confirmation that the operation has been performed correctly. This can result in data corruption or render the transponder unusable if the Lock command has been used.

In a situation were more than one transponders are within the reader's range and the transponders (i.e. SAMPT) supports addressed commands, addressed operations can be used. Before an addressed operation can be performed, the selective address of the transponder should be known to the reader, otherwise the transponder cannot be accessed. The selective address consists of the 3 least significant bytes of the data of page 1 of the transponder. This page can be read by using the available read functions (see chapter 'Read functions'). If multiple transponders are within range of the reader during the addressed operation, only the addressed transponder will perform the operation and respond to the reader. This prevents that collisions occur during the operation.

Page locking

A lockable page of a SAMPT or a Multipage transponder can only be locked once. A locked page cannot be modified by any subsequent command. It is permanently locked and the data contained in the page cannot be changed.
Lockable pages can be locked using the library functions provided for this purpose.