CZECH
TECHNICAL
UNIVERSITY
IN PRAGUE

Faculty of Information Technology

Smart re-use of hardware peripherals for better software UART

Jan Bělohoubek

jan.belohoubek@fit.cvut.cz
Czech Technical University in Prague
Faculty of Information Technology

The 3rd Prague Embedded Systems Workshop

This work was in part supported by CTU grant SGS15/119/OHK3/1T/18

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

New feature/peripheral
HW/SW implementation comparison

- HW implementation
  - + higher performance
  - + performance of the original circuit is not (very much) influenced
  - – total cost
  - – time to market
  - ? additional static power consumption and dynamic power consumption is given by HW utilization

- SW implementation
  - – lower performance
  - – available CPU computation time reduced
  - + total cost
  - + time to market
  - ? power consumption given (not only) by software execution time

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

Technology characteristics

Low-power microcontroller

- Microcontroller characteristics:
  - → 8-bit RISC architecture CoolRisc core
  - → up to 5 MIPS at 10 MHz
  - → limited abilities for "universal" communication – GPIO, configurable 8-bit SPI master/slave

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

HW or SW implementation?

- General GPIO implementation limitations:
  - → precise timing is not well achievable in software, when another asynchronous events are considered
  - → interrupt-driven implementation interrupts the CPU core frequently
- The presented software implementation of UART extensively uses HW dedicated for SPI
  - → fast implementation
  - → relatively high performance
  - → low impact to CPU performance (and power consumption)

CZECH
TECHNICAL
UNIVERSITY
IN PRAGUE

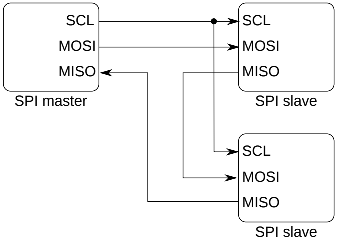
Faculty of Information Technology

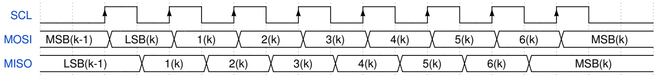
Well-known technologies

SPI overview

- 3-wire interface – SCL, MOSI, MISO (and possibly SS – slave select signal)
- single-master, multi-slave
- synchronous communication

SPI – daisy-chain



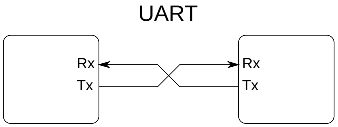


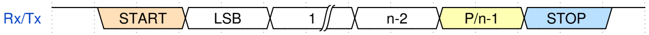
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

Well-known technologies
UART overview

- 2-wire interface – RxD, TxD
- point-to point connections
- asynchronous communication
- debugging, superior system interfacing





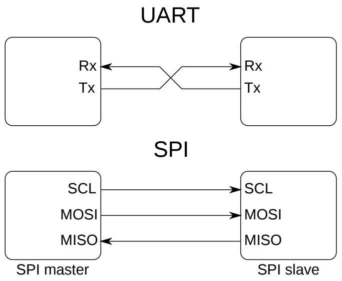
CZECH TECHNICAL UNIVERSITY IN PRAGUE

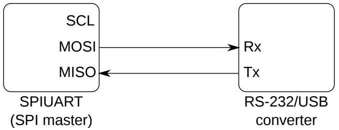
Faculty of Information Technology

SPIUART implementation

The Idea

- similar interfaces – duplex communication
- UART device is always a communication initiator
- SPI master can also be a communication initiator



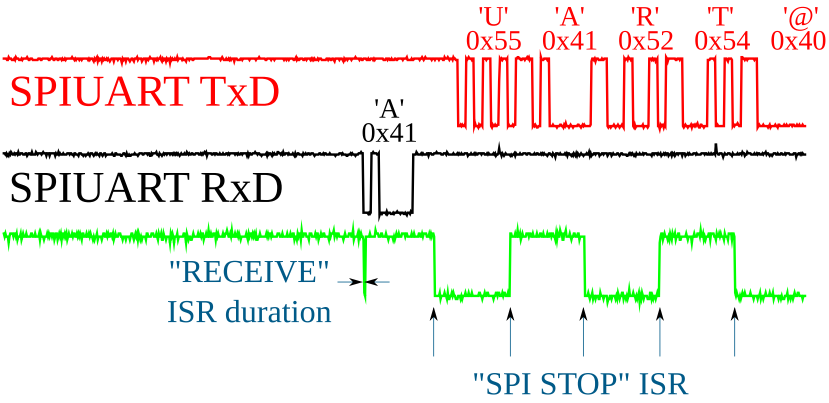


SPIUART implementation
Requirements and limitations

- SPI master
  - → ability to initialize SPI during START BIT reception limits the baud rate
  - → SCL pin is occupied even if it is not required for communication
  - → SPI bit width limits the UART data bit count
    - transmission (SPI master) – START, STOP and 6 DATA bits
    - reception (SPI master) – START + 7 DATA bits (possibly 8 DATA bits)

SPIUART implementation

Measurement

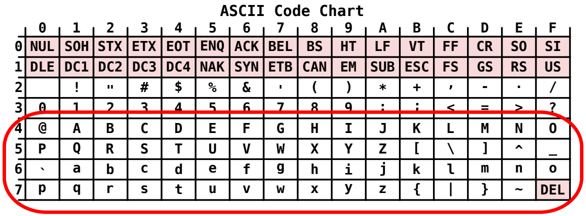


Faculty of Information Technology

SPIUART – trade-off choice

UART DATA bit count selection

- asymmetric 6/8 DATA bit UART can be implemented
- to increase the effectiveness, 8 should be selected $\rightarrow$ two MSBs received from SPI master are always equal to STOP bit (one)
- if 7 is selected, ASCII terminal can be used for communication and debugging


ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

0x40 ... 0x7F

An image from Wikipedia, The Free Encyclopedia, www.wikipedia.org

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

Results

Comparison with existing GPIO implementation

	EM68xx softUART	SPIUART
Transmission data bit count	8	6
Reception data bit count	8	8
Baud rate [bit/s]	fixed 2400	up to 9600
# ISRs executed during transmission	9	1
# ISRs executed during transmission/data bit	1,125	0,167
# ISRs executed during reception	10	2
# ISRs executed during reception/data bit	1,250	0,250
C code including preprocessor directives	up to 140 lines	up to 250
Assembly language equivalent	240 lines	127 lines
Number of instructions	972	504
Used program memory	6%	3%
Number of *NOP* instructions	87	0
Instructions in ISRs	656	268
The share of instructions in ISRs	67%	53%
Used I/O PINs	2	3