COPYRIGHT

DISCLAIMER

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245 - 2020

Lecture 16

Serial Communication – UART/
Seriële Kommunikasie – UART

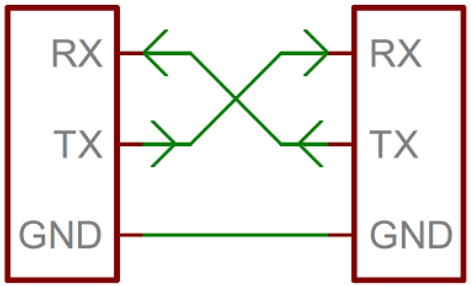
Dr Rensu Theart & Dr Lourens Visagie

Standard Serial Communication
Standaard Seriële Kommunikasie

	Half/Full-duplex	Bus/point-to-point	Synchronous/ Asynchronous
UART (Universal Asynchronous Receiver/Transmitter	Full	Point-to-point	Asynchronous
I^2C (Inter-Integrated Circuit)	Half	Bus	Synchronous
SPI (Serial Peripheral Interface)	Full	Bus	Synchronous

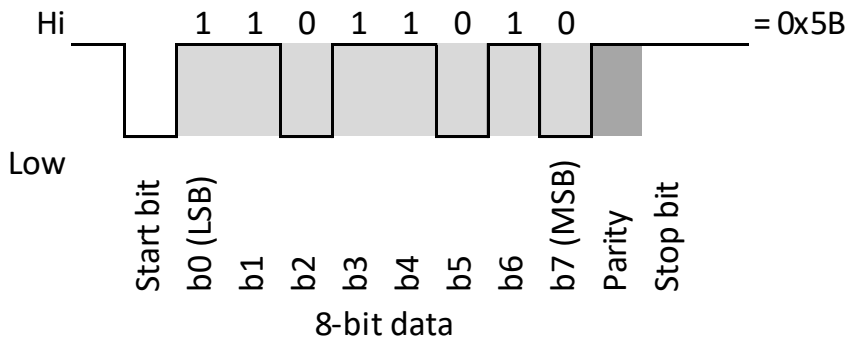
UART

- UART = Universal Asynchronous Receiver/Transmitter
- Asynchronous – no clock signal
- Point-to-point – connects two devices with each other
  - Specifically, a full-duplex point-to-point link (data can be sent both ways at the same time)
- Microcontroller signal levels : logical 0 = 0V, logical 1 = 3.3V or 5V
- Three signals: TX, RX and Ground
- Both transmitter and receiver must use the same bitrate (baud rate) – some tolerance is allowed (specification in datasheet)
- there are some standard bitrates - 1200, 9600, 19200, 115200 bps



UART

- Idle state is high voltage (3.3V or 5V)
- Data is transmitted in a frame
  - Start bit (low)
  - Data bits (usually 8 bits, but can be configured)
  - Optional parity bit
  - One or two stop bits (high)



- Different configuration options: baudrate, number of data bits, parity included (even or odd parity), number of stop bits
- 8N1 = 8 bits data, no parity bit and one stop bit (common configuration)
- Data is transmitted least-significant bit first

UART

Signal bit-rate, and data transfer rate

- With 8N1 configuration (8 data bits, 1 start bit and 1 stop bit) we are transferring 10 bits to convey 8 bits (1 byte) of useful information
- Thus if the UART signal bit-rate is 115200 bits per second, using 8N1, the data transfer rate is 11520 bytes per second

RS-232

- "Older" standard, not used much anymore
- Different signal levels (+12V and -12V)
- Need external hardware (RS-232 driver IC) connected between microcontroller and external communication lines

Other

- Additional signals for synchronization or hardware "Flow Control" (modem interface)
- Sometimes the hardware module is called a USART (Universal Synchronous Asynchronous Receiver/Transmitter) – Then it can also output a clock signal if you want to
- UART clock is usually driven by one of the on-board timer modules
- (but not on the STM32F4 – UARTs have their own clock signal generator)

U(S)ART on STM32F411
U(S)ART binne die STM32F411

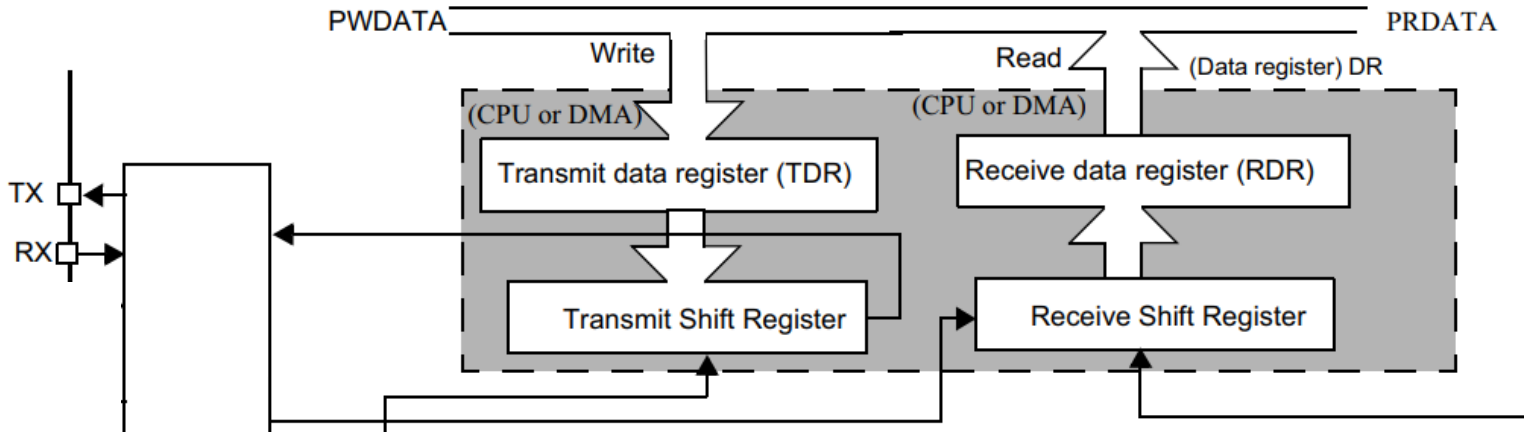
As with all embedded peripherals, interaction is through memory mapped registers

- USART_SR : Status Register
- USART_DR : Data Register
- USART_BRR : Baud Rate Register
- USART_CRx : Configuration Registers

Transmit and Receive makes use of shift registers

Received data is placed in the USART_DR. Data written to the USART_DR gets transmitted through shift registers

Figure 167. USART block diagram

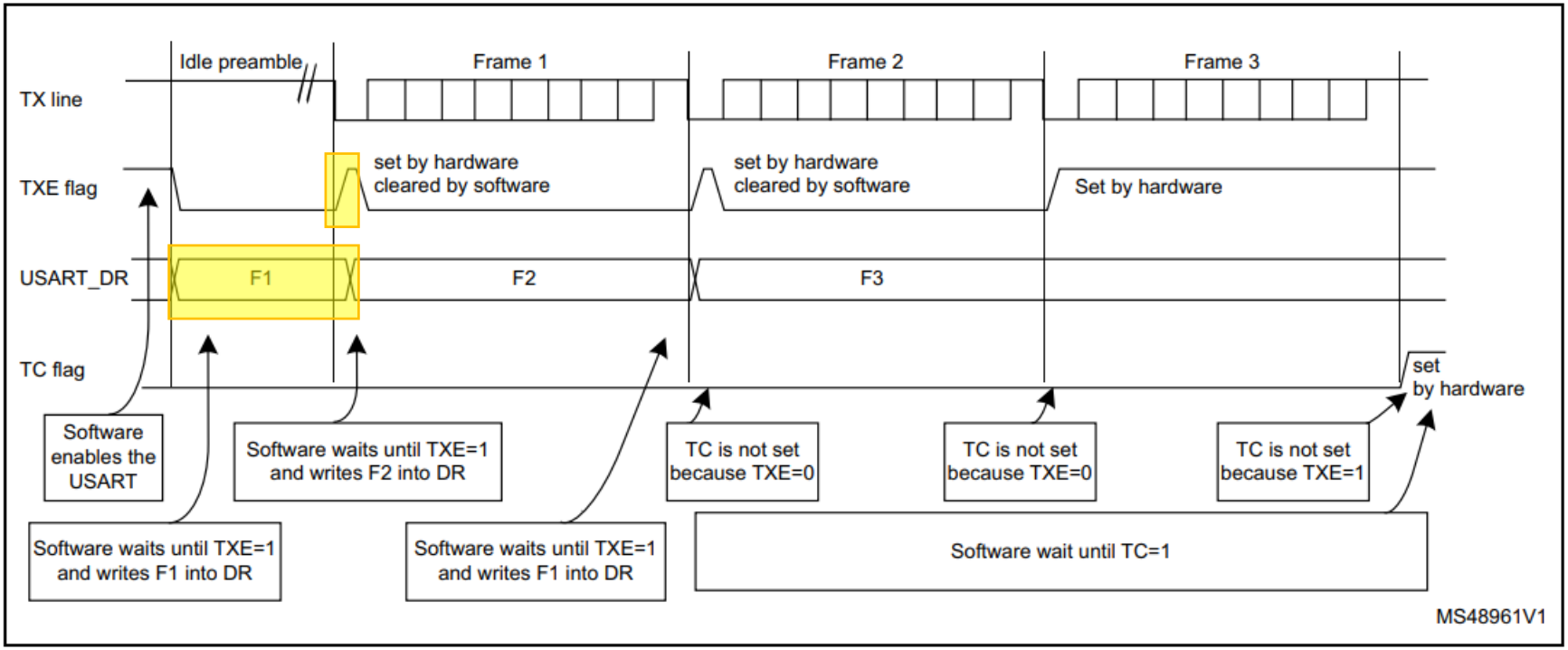


U(S)ART on STM32F411: Data Transmission
U(S)ART binne die STM32F411: Data Versending

Transmission starts by writing data into the DR (data register)

Data is moved into the transmit shift register, and the TXE (Transmit Data Register Empty) flag will go high - indicating that it is ok to write more data into the data register (DR)

Figure 170. TC/TXE behavior when transmitting

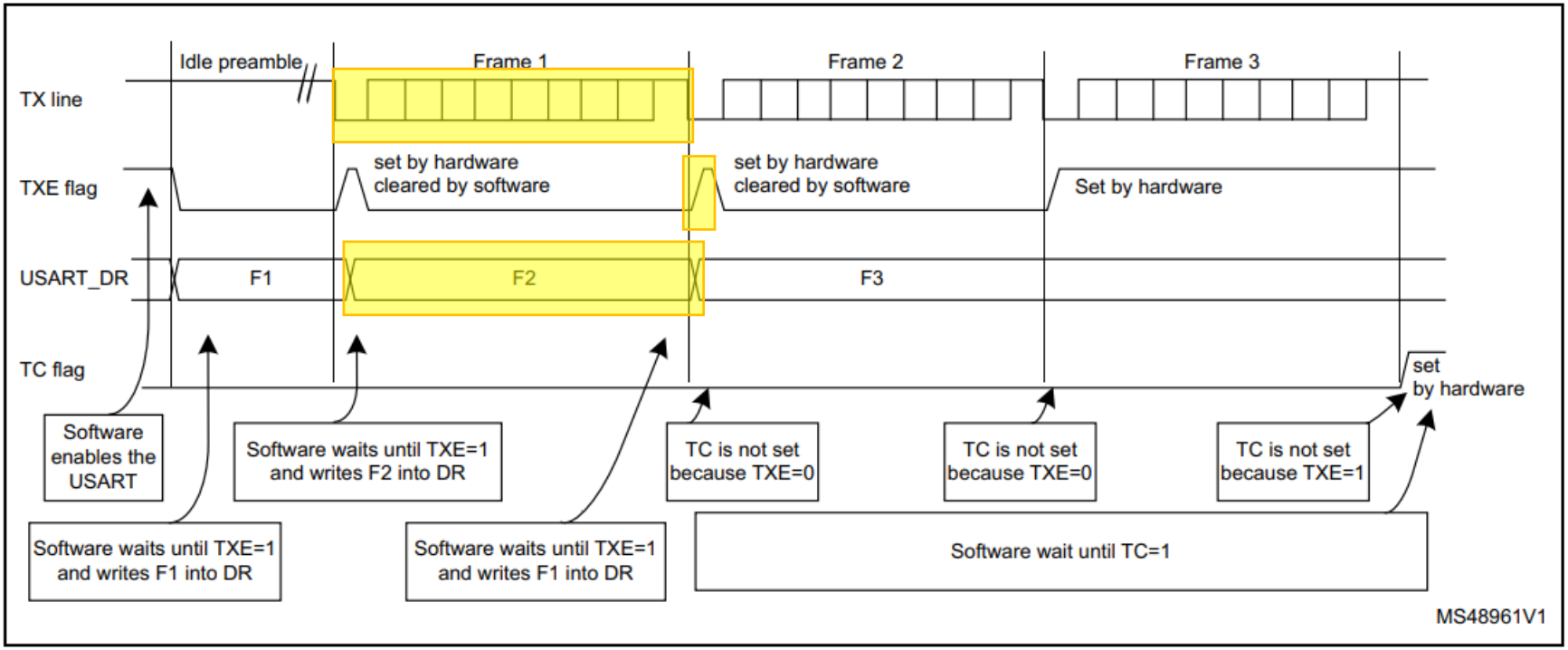


U(S)ART on STM32F411: Data Transmission
U(S)ART binne die STM32F411: Data Versending

If there is more data to send, clear the TXE flag in software and write another data frame to the DR.

While the first data frame is still being transmitted out of the shift register, the contents of the DR cannot be copied. Wait until the TXE flag goes high again.

Figure 170. TC/TXE behavior when transmitting

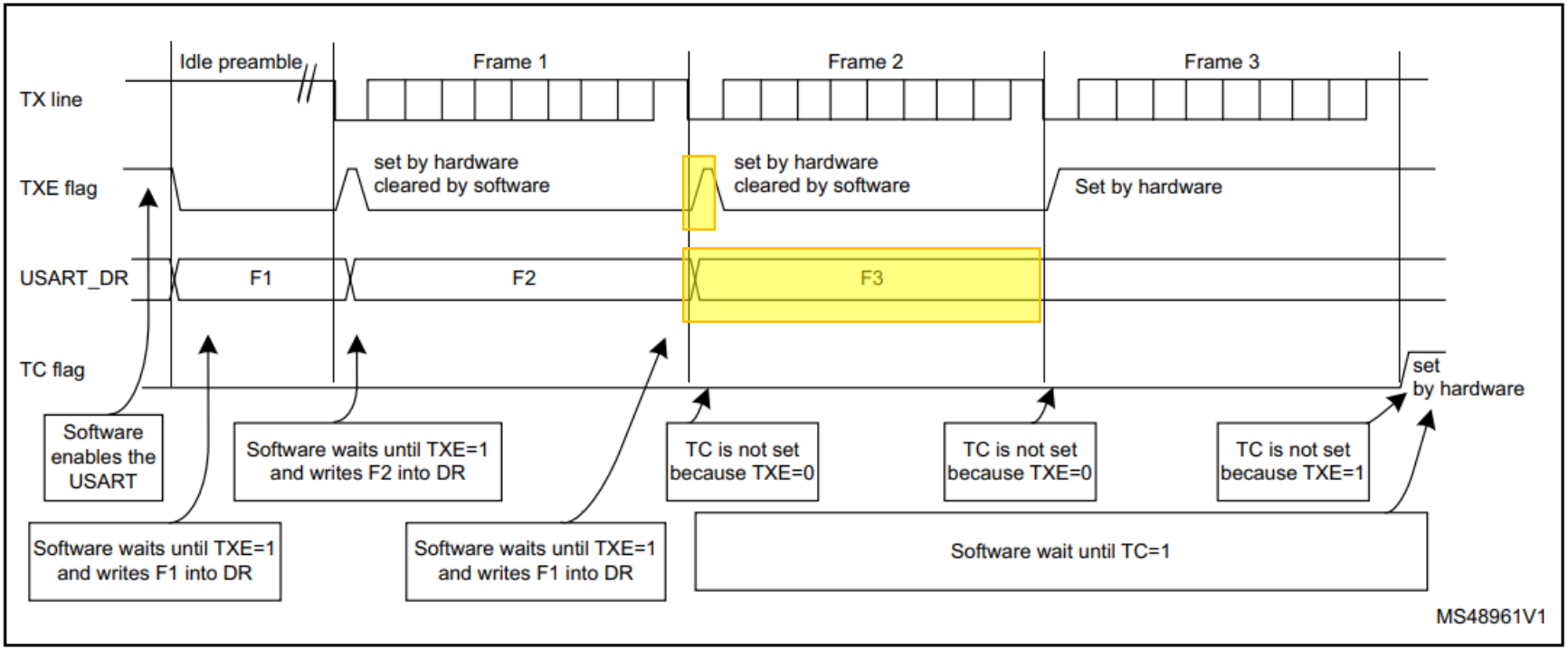


U(S)ART on STM32F411: Data Transmission
U(S)ART binne die STM32F411: Data Versending

When the shift register is empty, the pending data in the DR will automatically be transferred into the shift register and the TXE flag will again go high.

If there is still data to send, write to the DR

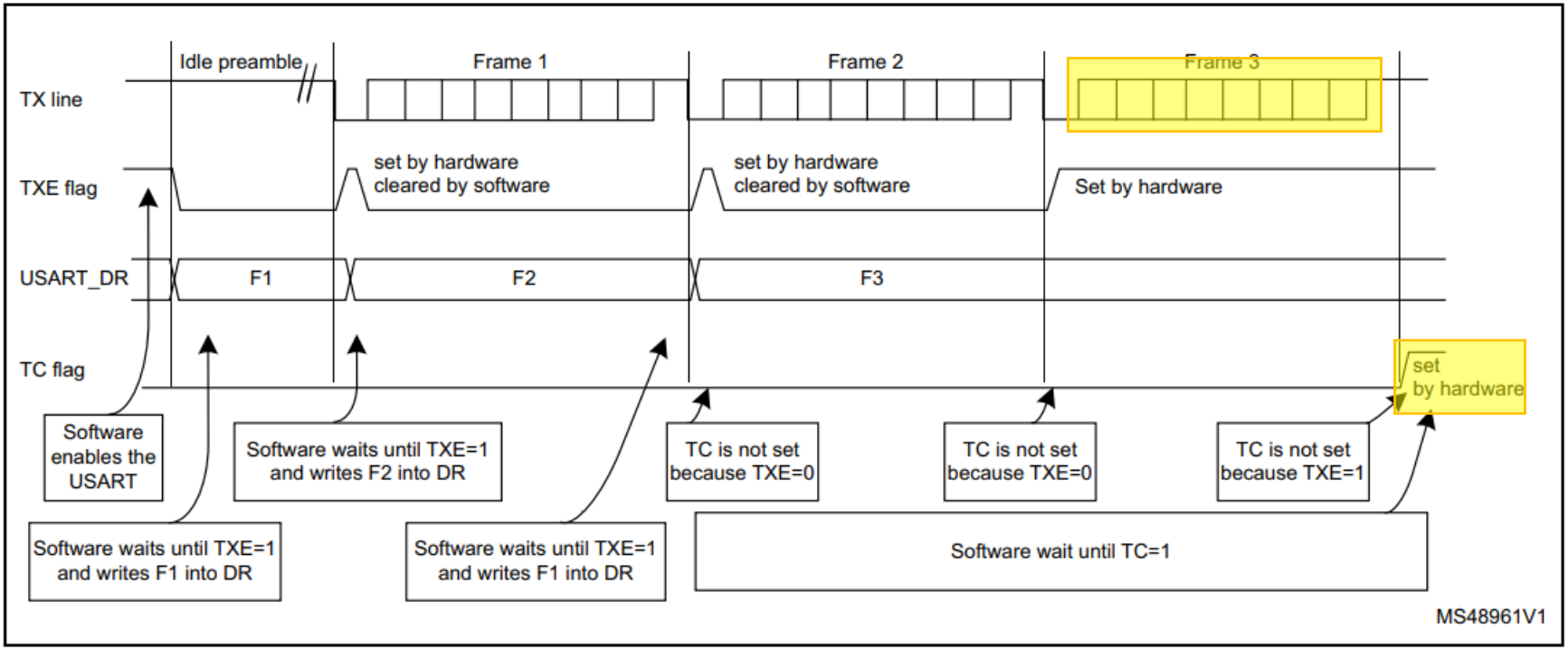
Figure 170. TC/TXE behavior when transmitting



U(S)ART on STM32F411: Data Transmission
U(S)ART binne die STM32F411: Data Versending

Process continues until there is no more data to send. Once the shift register is empty (last frame has finished transmitting) and the TXE bit is still set (there is no pending data frame into the DR), the TC (Transmit Complete) flag will go high

Figure 170. TC/TXE behavior when transmitting



U(S)ART on STM32F411: Data Reception
U(S)ART binne die STM32F411: Data Ontvangs

- Receiver makes use of oversampling for noise rejection and improved tolerance to clock deviation:
  - Either 8x or 16x over sampling
  - Received signal is sampled 16x faster than expected baud rate
  - Detect start bit from oversampled signal
- Upon detection of start bit, following bits are clocked into the receiver shift register
- When the entire frame has been received, it is copied to the receive Data Register (DR) and the Receive Data Register Not Empty (RXNE) flag goes high
- Software reads from the DR to clear the RXNE flag.
- If another data frame is received while the RXNE flag is still high, an overrun error occurs (ORE flag is set)

U(S)ART on STM32F411: Baud Rate
U(S)ART binne die STM32F411: Simbool Tempo

- Baud Rate for both receive and transmit is controlled through the USARTDIV value in the BRR (Baud Rate Register)
- Input clock for USART1 and USART6 comes from APB2 clock
- Input clock for USART2 comes from APB1 clock
- For 8x oversampling:
  - $Baud\ rate = \frac{f_{ck}}{8 \times USARTDIV}$
- For 16x oversampling:
  - $Baud\ rate = \frac{f_{ck}}{16 \times USARTDIV}$
- USARTDIV is an unsigned fixed point number, coded in the USART_BRR register

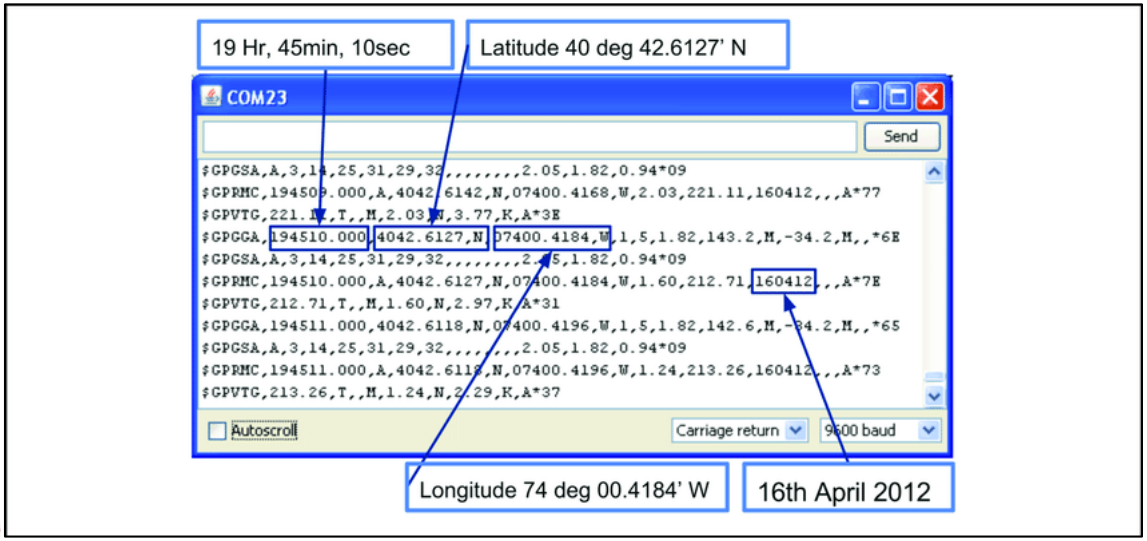
U(S)ART on STM32F411: Interrupts
U(S)ART binne die STM32F411: Onderbrekings

Table 86. USART interrupt requests

Interrupt event	Event flag	Enable control bit
Transmit Data Register Empty	TXE	TXEIE
CTS flag	CTS	CTSIE
Transmission Complete	TC	TCIE
Received Data Ready to be Read	RXNE	RXNEIE
Overrun Error Detected	ORE	
Idle Line Detected	IDLE	IDLEIE
Parity Error	PE	PEIE
Break Flag	LBD	LBDIE
Noise Flag, Overrun error and Framing Error in multibuffer communication	NF or ORE or FE	EIE

UART application examples
UART toepassings voorbeelde

- GNSS (Global Navigation Satellite System) Receiver (GPS)
- GNSS module includes all functionality to
  - Receive signals from satellites
  - Compute position, velocity and time
- UART communications interface
  - Microcontroller can adjust setup of GNSS receiver
  - Receiver outputs periodic navigation solution
- Output messages make use of GPS NMEA format

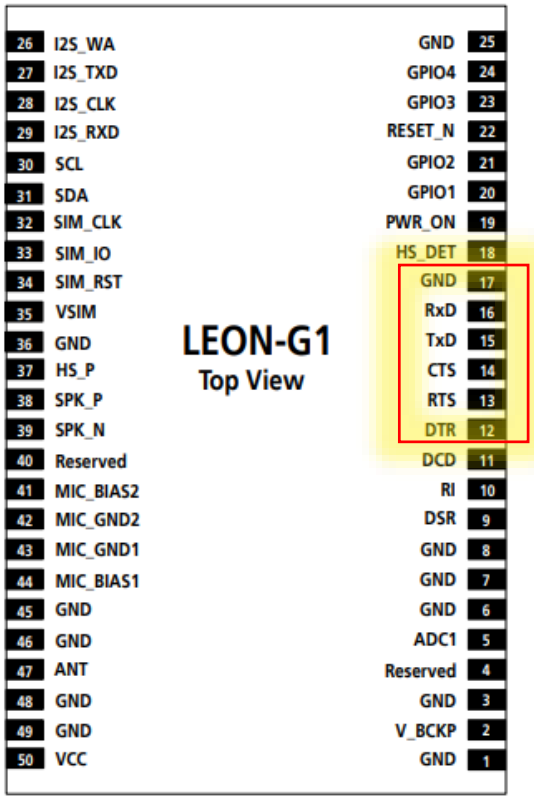


GPS, Adafruit ver 3.0 module

Specifications:
Module specs:
- Satellites: 22 tracking, 66 searching
- Patch Antenna Size: 15mm x 15mm x 4mm
- Update rate: 1 to 10 Hz
- Position Accuracy: 1.8 meters
- Velocity Accuracy: 0.1 meters/s
- Warm/cold start: 34 seconds
- Acquisition sensitivity: -145 dBm
- Tracking sensitivity: -165 dBm
- Maximum Velocity: 515m/s
- Vin range: 3.0-5.5VDC
- MTK3339 Operating current: 25mA tracking, 20 mA current draw during navigation
- **Output: NMEA 0183, 9600 baud default**
- DGPS/WAAS/EGNOS supported
- FCC E911 compliance and AGPS support (Offline mode : EPO valid up to 14 days)
- Up to 210 PRN channels
- Jammer detection and reduction
- Multi-path detection and compensation

UART application examples
UART toepassings voorbeelde

- GSM (Cellular) modem
- Send and receive data from microcontroller to cellular network
  - Internet data connection
  - IoT (Internet of Things) applications
  - Send SMSs
- Complicated protocol on UART communications



ublox u-blox Cellular Modules - AT Commands Manual

7.26 Provide cell information +UCELLINFO

+UCELLINFO						
Modules	LARA-R2 TOBY-R2					
	SARA-U2 LISA-U2 LISA-U1					
Attributes	Syntax	PIN required	Settings saved	Can be aborted	Response time	Error reference
	partial	No	No	No	*< 5 s*	*+CME Error*

7.26.1 Description

Provides some information about the serving and neighbour cells (for 2G RAT, 3G RAT and 4G RAT). This information can be:

- Periodic: it is performed enabling the URCs through the set command. If enabled, the URCs are periodically issued providing the status about the serving and neighbour cells.
- One-shot: it is performed issuing the read command.

7.26.2 Syntax

Type	Syntax	Response	Example
Set	AT+UCELLINFO=<mode>	OK	AT+UCELLINFO=1 OK
Read	AT+UCELLINFO?	**2G cells:** +UCELLINFO: <mode>,<type>,<MCC>,<MNC>,<LAC>,<CI>,<RxLev>[,<t_adv>[,<ch_type>,<ch_mode>]]	+UCELLINFO: 0,0,222,1,D5BD,5265,36,1,255,255 OK