Sander Vermolen   Hugo Brakman Vincent Driessen
Joseph Kavuma Laura Nij Bijvank

Supporting Formal Method Teaching with Real-Life Protocols

Radboud University Nijmegen

Outline

- Solution outline

- Bluetooth example
  - Informal description
  - UPPAAL model

- Results using this solution

- Required knowledge and skills

- Variations (within solution outline)

Solution outline

Real-Life protocol
- Well known
- Protocol specification available

Abstract model
- UPPAAL
- PRISM

Verification
- Correctness
- Timing issues

Bluetooth project

Real-Life protocol

Bluetooth

Abstract model

UPPAAL

Verification

Devices will eventually communicate

(No deadlock)

Analysis of Embedded Systems
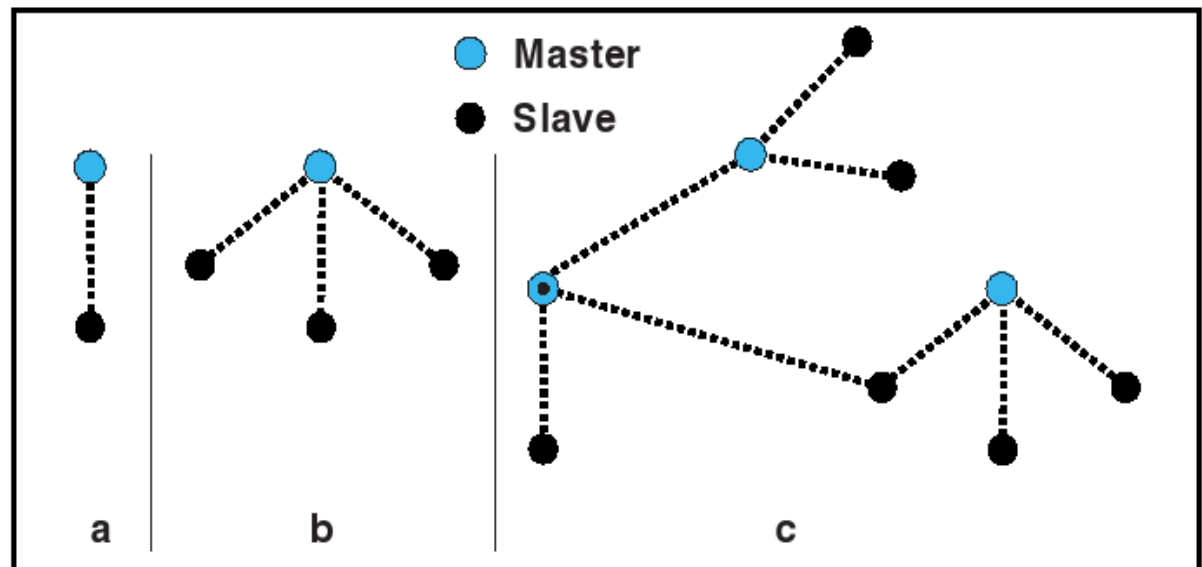
Prof. dr. Frits W. Vaandrager

Informal description

Inquiry phase

Master-Slave relation
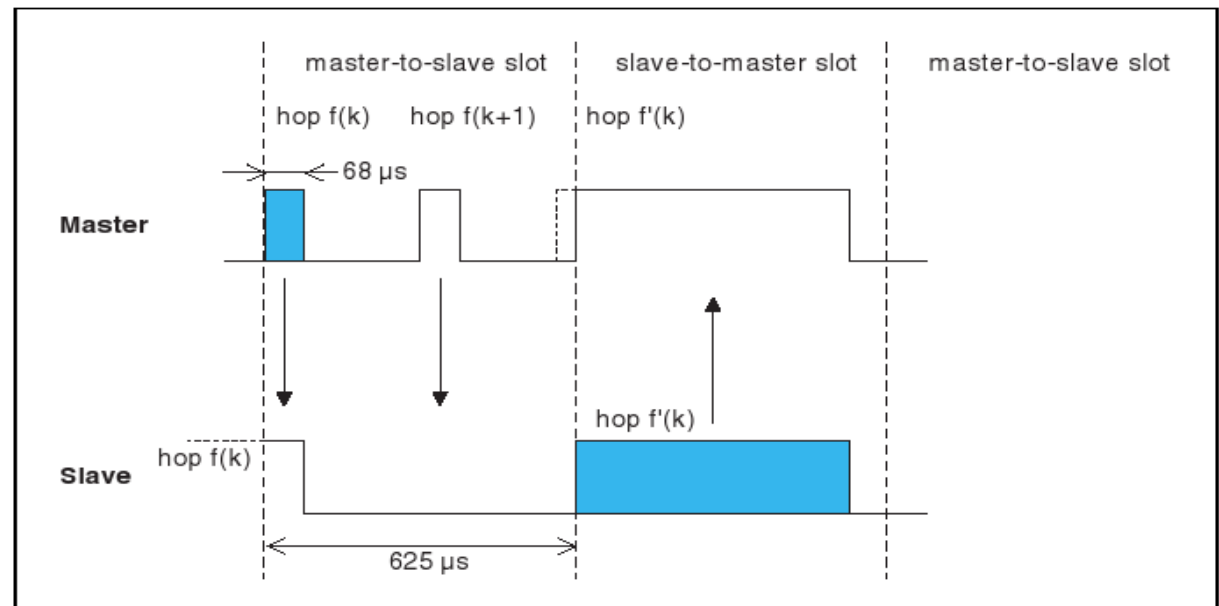
Hopping

Communication scheme

TT / RR Communication

Hop on each slot
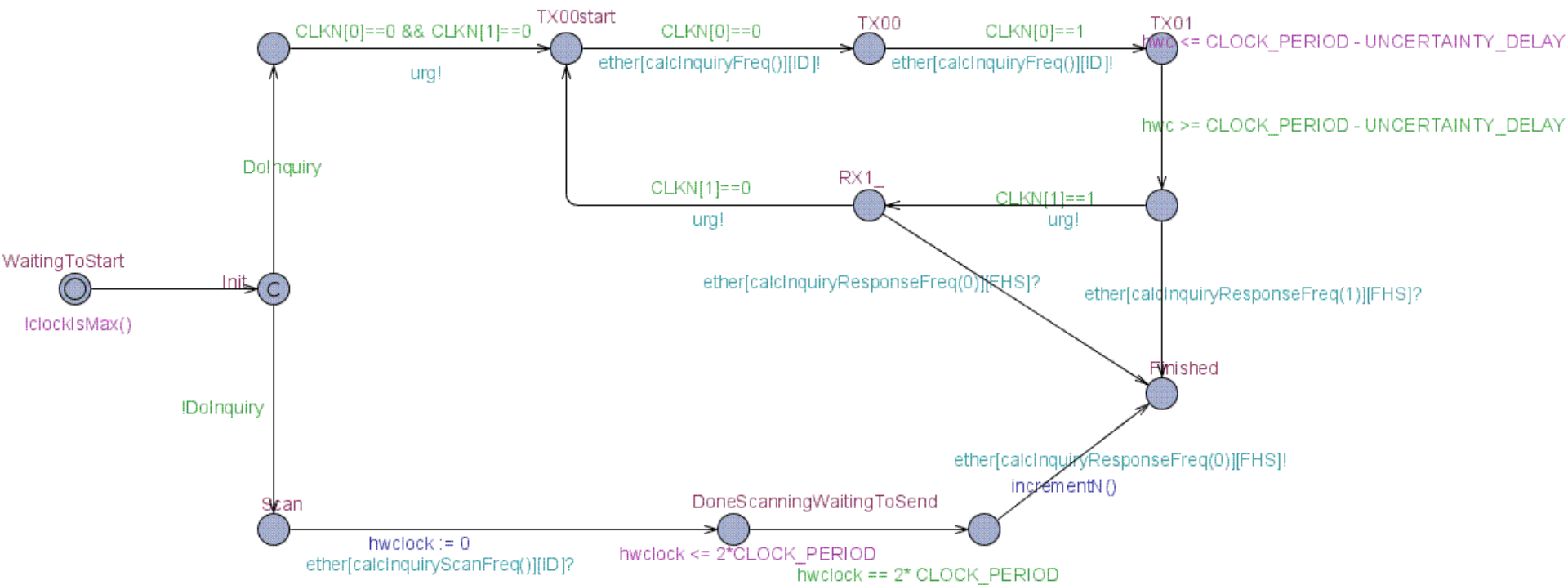
Clocks not synchronized

Jitter and Drift

UPPAAL Model – Top level

```
MasterBTClock      := BluetoothClock(CLKNs[MASTER], hwclocks[MASTER]);
SlaveBTClock       := BluetoothClock(CLKNs[SLAVE], hwclocks[SLAVE]);
Master             := Device(MASTER, CLKNs[MASTER], hwclocks[MASTER]);
Slave              := Device(SLAVE, CLKNs[SLAVE], hwclocks[SLAVE]);


system             Master, Slave, MasterBTClock, SlaveBTClock;
```
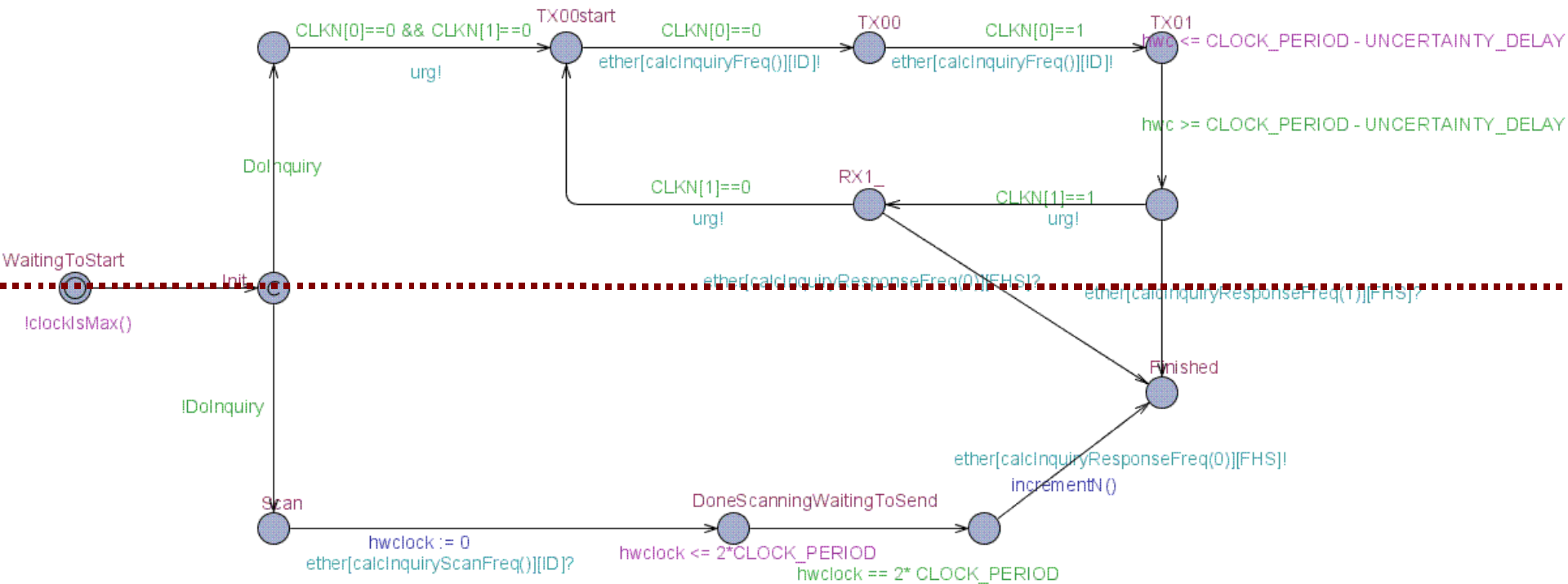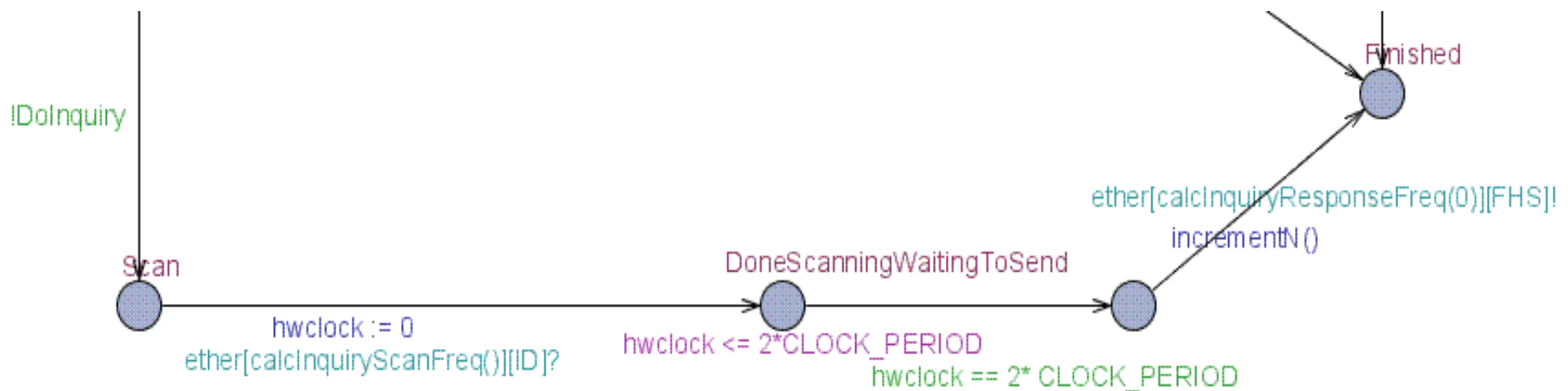
UPPAAL Model – Device

UPPAAL Model - Device

UPPAAL Model – Device – Slave

UPPAAL Model – Device – Slave

!DoInquiry

Finished

ether[calcInquiryResponseFreq(0)][FHS]!
incrementN()

Scan

DoneScanningWaitingToSend

hwclock := 0
ether[calcInquiryScanFreq()][ID]?

hwclock <= 2*CLOCK_PERIOD

hwclock == 2* CLOCK_PERIOD
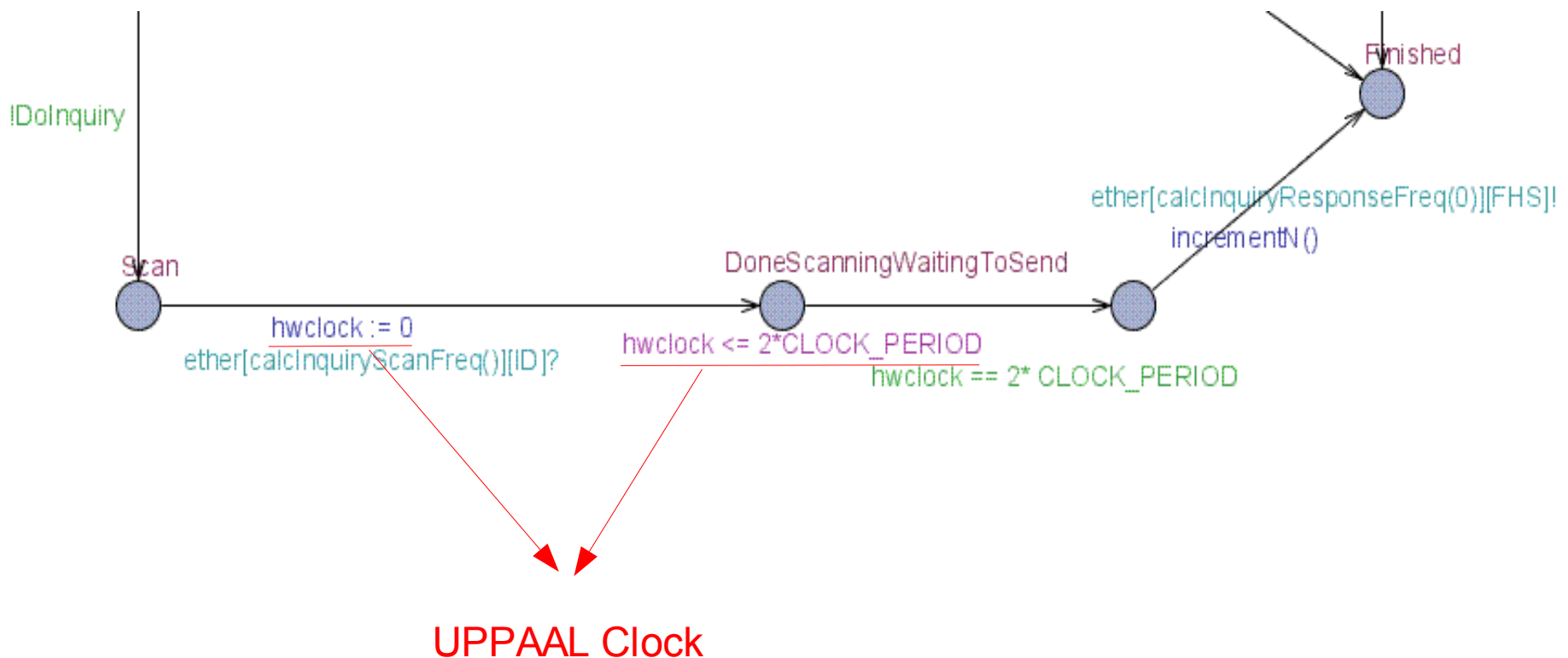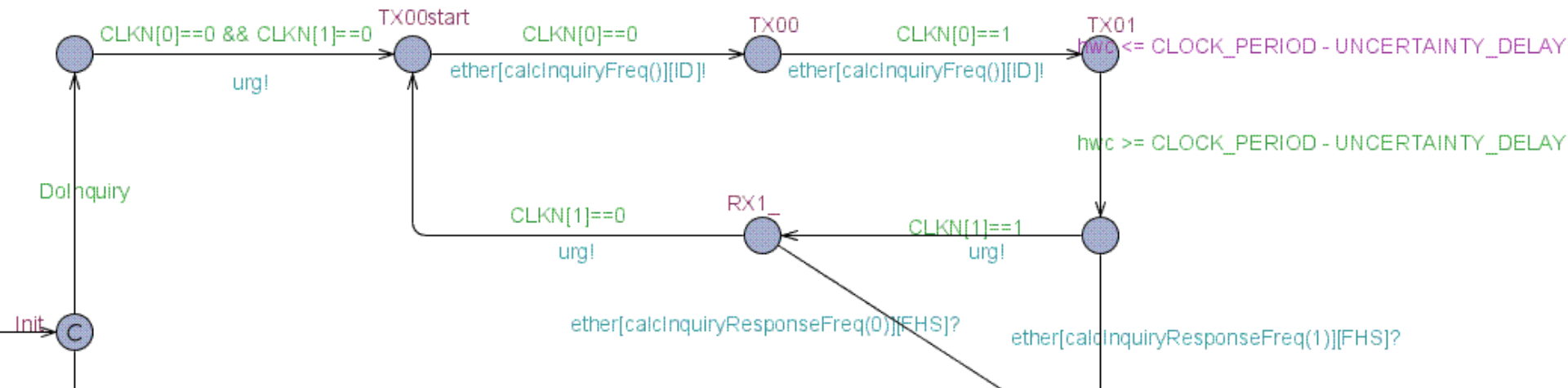
UPPAAL Clock

UPPAAL Model – Device – Master

UPPAAL Model - Code

- Hopping sequence calculation
- Clock counter
- Constant Declarations

Project results

- For all possible traces:
    - All devices will eventually be finished
    - Deadlock will never occur
- Unclear details of Bluetooth protocol document

- Including clock jitter and drift
    - Jitter of   6/625
    - Drift of   1/625

- Reduced clock counter
- Transmissions are instantaneous

UPPAAL

Pro's

- Intuitive and user-friendly interface
- Drawing automata increases the overview
- Very good feedback through a graphical simulator

UPPAAL

Cons

- Not object oriented
- Limited programming capabilities
- Poor verification errors
- Some small bugs

Practical issues

- Group size
- Duration

Results

- Realistic
  - Improves motivation
  - Recognise the values of the teaching

- Graphical visualisation/simulation

- Group work helps weaker students

- Actual results motivate the student

Required knowledge and skills

- Mathematical background
- Understanding of tools and theories
- Team work

Variations within the solution outline

Variations within Bluetooth protocol

- Duration of transmission
- More network nodes or merging networks
- Other phases
- Extending the model

Different protocols

Different approaches (tools)

- Probabilistic