Oberseminar
Wintersemester 2004

Stephan Höhrmann

Christian-Albrechts Universität Kiel
Department of Computer Science and Applied Mathematics
Real-Time Systems and Embedded Systems Group

14.12.2004

An Ultrasonic Positioning System for Lego Mindstorms Robots

Motivation

- Most autonomous mobile robots need to know their position to fulfill their tasks
- With the standard Mindstorms sensors there is just one possibility: *dead reckoning*
  - Many sensors needed for good results
  - Requires permanent computations
  - Error grows with time and without boundaries
  - Several sources of error (battery voltage, friction, motor characteristics, measurement errors,...)
- Very unreliable
- No absolute position (independent of previous movements or measurements)

Determining absolute positions

The goal of this work is to build and evaluate a positioning system for Mindstorms robots

- An area is prepared by placing beacons at known locations
- The Mindstorms robot can move freely within the area
- Distance to the beacons is measured by the time of flight of ultrasonic bursts between the beacon and the robot
- The system can measure relative and absolute times if an additional signal is used
- The robot's position can be calculated from the measured times and the beacon geometry

An ultrasonic positioning system

A number of steps was needed to build the system

- Understand Mindstorms electronics well enough to be able to build compatible hardware
- Learn to program and use microcontrollers as they are needed to control low level operations
- Create and detect ultrasonic bursts
- Design of sender and receiver circuitry
- Design of a system to control and synchronize the beacons
- Develop algorithms to calculate the position
- Write software for the whole system (Assembler + C), including brickOS interfaces

Advances since February (1/2)

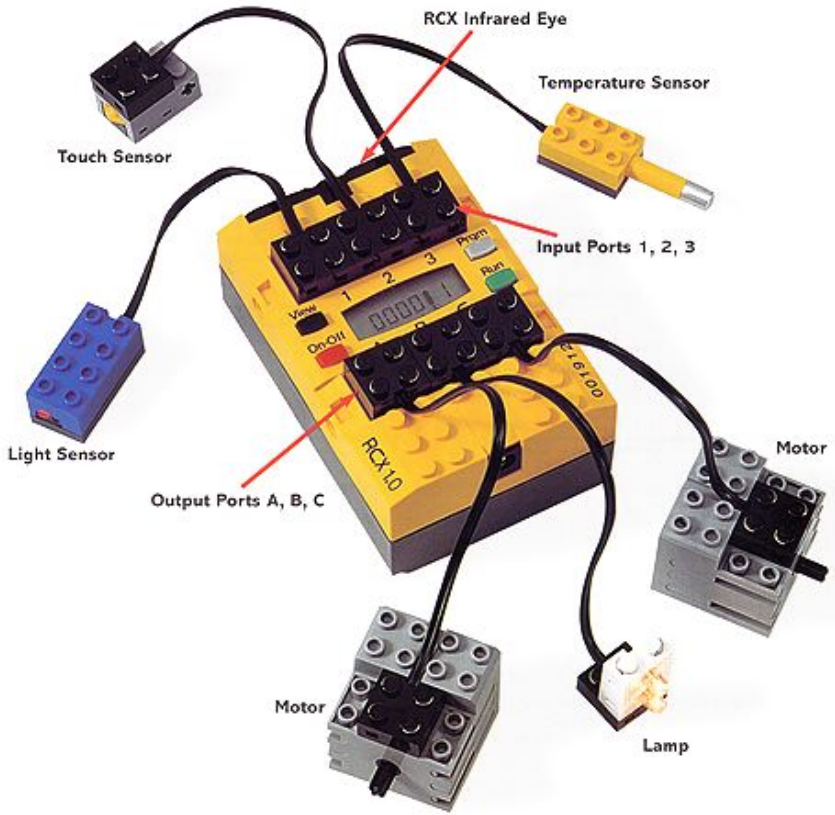
- Extension of microcontroller equipment
  - New prototyping boards (16F628)
  - Debug interface, automated testing possibility
- Major design changes in both sender and receiver
  - Reduced power consumption from 15 mA to 4 mA
  - Optimized both sending intensity and receiver sensitivity, range extended from 3 m to 10 m while using reflectors
  - Simplified circuit to use less parts and space
  - Moved from infrared to RF communication at 433 MHz
  - Beacons now identify themselves via RF
  - The sensor sends more detailed data to the RCX (time of flight, time since last burst, duration of echo, flags to detect receiving errors)
  - Dramatically decreased distance measuring errors from 20 mm to 0.35 mm (one microsecond)

Advances since February (2/2)

- Design and construction of beacon controlling system
  - Up to 4 beacons can be connected to a master in a daisy chain
  - The system (re-)configures automatically
  - User interface to view system status and change sender timing
  - Timing error in the system is limited to a microsecond
- Numerical simulation of errors in position calculation
- Calculation of good reflector geometry
- Improvements of whole system software
  - Port of software and kernel patches to brickOS 0.2.6.10
  - Source code documentation added
  - Added user level interfaces to brickOS
  - Increased robustness and error tolerance, added error recovery
  - Removal of all remaining bugs?
- Built many printed circuit boards, cases, reflectors,...

Overview

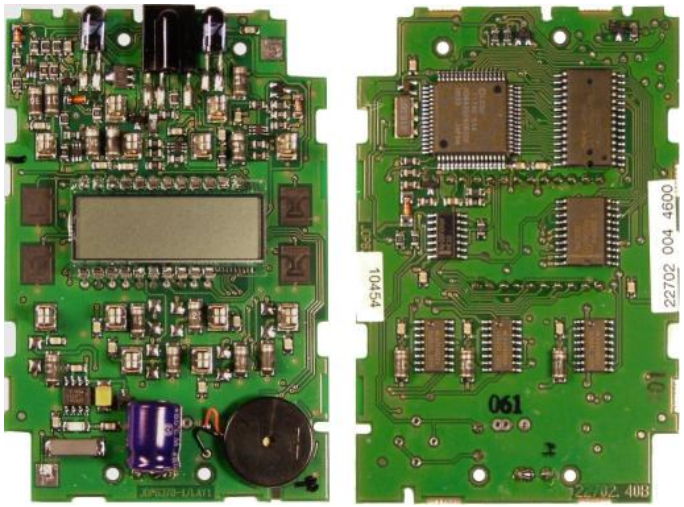
Lego Mindstorms electronics



- Embedded system with drivers for actuators...
  - Motors
  - lamps
- ...and sensors
  - Touch sensor
  - light sensor
  - Rotation sensor
  - Temperature sensor
- Programmable using cross compilers

Mindstorms internals

Construction of peripherals requires knowledge about driver characteristics and operations



- Many basics were uncovered by Proudfoot, Nelson and Gasperi
- Not sufficient, additional reverse engineering was needed
- Focus on actuators, sensors and infrared communication

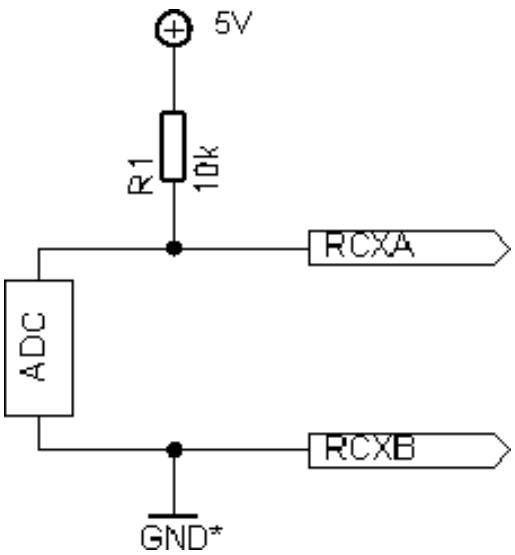
Actuator drivers

- The RCX contains 3 programmable actuator drivers
- Any of them can operate in one of the modes
  - forward
  - Reverse
  - off
  - brake
- Speed can be set by pulse width modulation
- pulse width modulation can also be used to send information to actuators
- Outputs are short-circuit protected
- Current is limited to 500 mA

Sensor basics

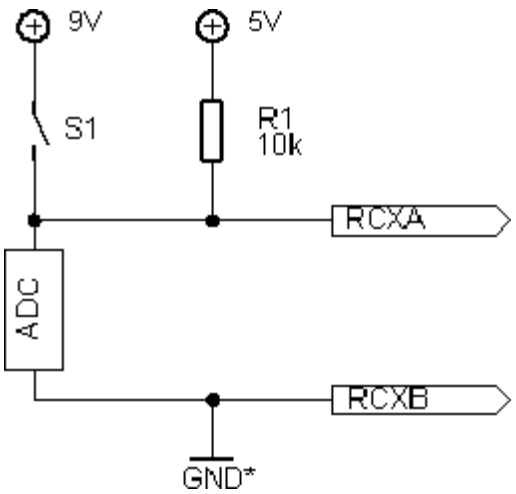
- 4 sensors are supported (3 external + battery sensor)
- Resolution is 10 bits (0..1023)
- Update frequency depends on firmware, 0.3 to 7 kHz
- Passive sensors are just resistors
  - Touch sensors (switch)
  - Temperature sensor (PTC)
  - Photo resistor
  - ...
- Active sensors get an additional power supply
  - Light sensor (LED is powered)
  - Rotation sensor (integrated electronics)
  - ...
- Sensor reading and power supply are multiplexed over cable with two wires

Passive sensors



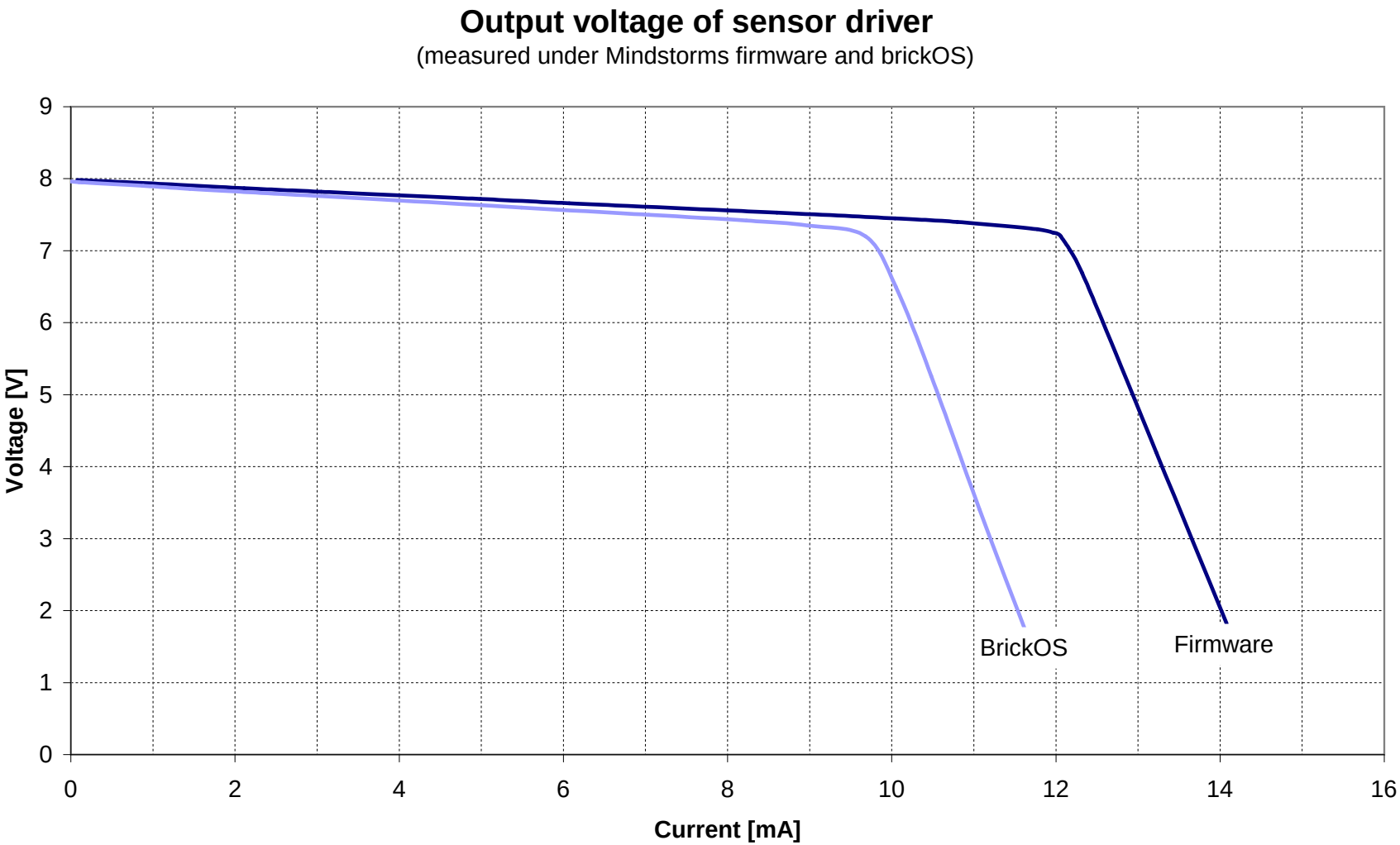
- ADC measures voltage across sensor
- Current through the sensor controls the sensor value
- Sensor reading is proportional to the current, not the sensor's resistance
- Only suitable for non-powered sensors

Active sensors

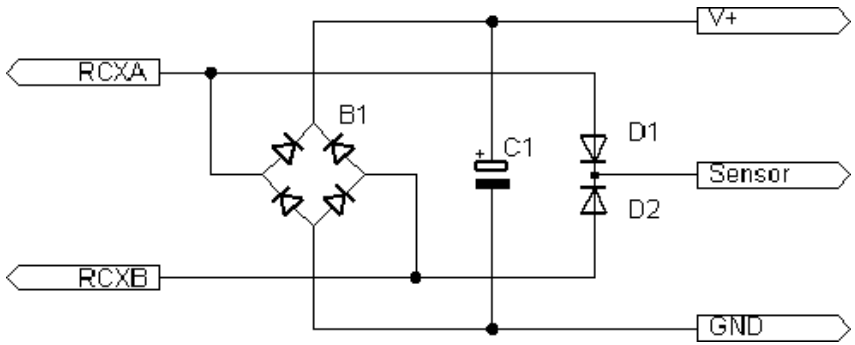


- Active sensors are sampled in the same way as passive sensors
- Sensor is powered between two readings
- Timing depends on the firmware
  - 3 ms supply + 0.1 ms reading (Lego)
  - 120 us + 40 us (brickOS)
- Sensor drivers are protected, current is limited to 12 mA

Sensor driver limitations



Building own sensors



- Bridge rectifier ensures correct polarity
- Capacitor buffers energy for short interruptions (sampling)
- Low power, low dropout, low quiescent current voltage regulator stabilizes output (if needed)
- Sensor reading current can only flow from the SENSOR connector to GND
- Any type of circuitry can be connected to this interface

Sound sensor

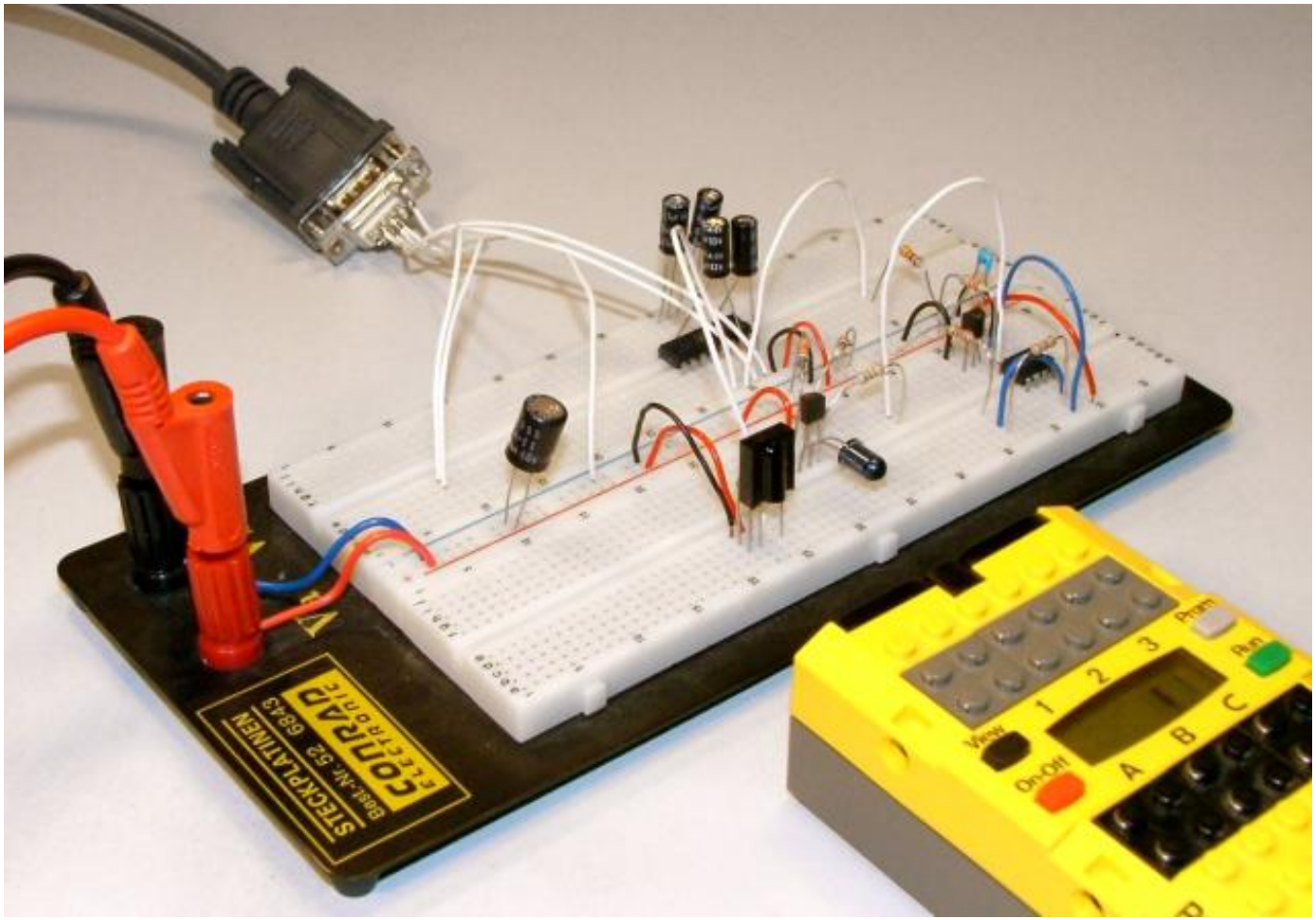
- First experiment in building active sensors
- Measures sound level
- Circuit by Michael Gasperi
- Can be built into Lego brick

This sensor was a good exercise in planning and building printed circuit boards.

Infrared basics

- Both RCX and tower are equipped with infrared transceivers, allowing communication between them
- Mainly used for firmware and program downloads
- Serial data stream (RS232, 8 data bits, 1 stop bit, odd parity) modulated on 38 kHz carrier signal
- Collisions handled by an algorithm similar to CSMA/CD
- Transceiver can easily be built into own hardware

Transceiver prototype



Infrared communication

- Early plans of positioning system included infrared communication
  - Beacons can identify themselves
  - Synchronize clocks of sender and receiver
  - Robots can query or change settings of the beacon network
- These ideas have been discarded
- Some usable fragments of this work remain
  - Complete documentation of all involved protocols and algorithms
  - Replacement for lnpd and liblnp, working with all legOS/brickOS versions, USB tower and all protocols
  - USB tower support of brickOS repaired

Using microcontrollers

Microcontrollers can dramatically enhance a sensor's capabilities

- Many electronic components can be saved, reduces power usage
- Behavior is determined by software instead of hardware
  - Adaptive algorithms are possible
  - Complicated signal processing is possible
  - Functionality can easily be changed without touching hardware
  - Errors are easy to fix
- Precise position for Mindstorms detection is impossible without microcontrollers

PIC 12F765 features

- Device overview
  - 32 kHz - 20 MHz, integrated 4 MHz oscillator
  - 1024 instructions, 128 bytes RAM + EEPROM
  - 6 highly configurable I/O pins
  - 10 bit, 4 channel ADC
  - Comparator, programmable reference voltage
  - 8 bit and 16 bit timer/counter
  - Watchdog, serial programming interface
  - Low power consumption, around 1 mA
- Disadvantages
  - Complicated architecture (memory is organized in banks, no high level instructions, only 1 register, 8 level hardware stack)
  - Complicated debugging
- Flash memory, easy to obtain, costs 3 EUR

Development equipment

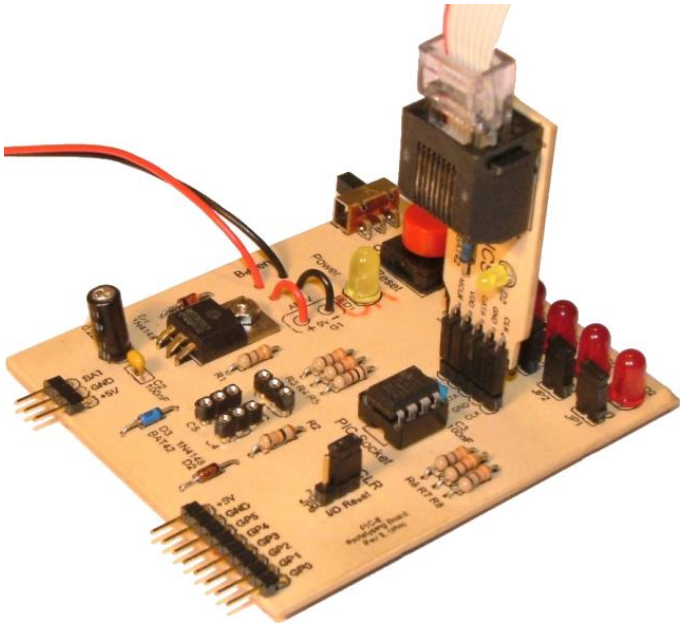
Software development for PIC microcontrollers is easy, tools are freely available

- Programs are written in assembler (high level languages are available, but their usability is low)
- A programmer device must be used to write the compiled binary image into the PIC's flash memory
- An application circuit is built around the PIC
- ICSP (in circuit serial programming) capabilities of both programmer and circuit are extremely useful
- The code must be tested and debugged, often inside the application circuit

PIC programmer

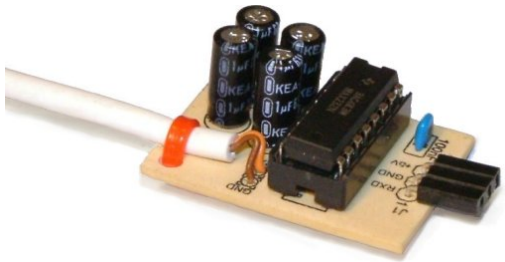
- Device developed by myself
- Can program all PICs via ICSP or built-in socket
- Specialized for ICSP usage, microcontroller can be programmed at run time
- Connects to parallel port
- Faster than other models
- ICSP interface is located on an external interface and thus variable

PIC prototyping board



- Prototyping boards reduce the time from an idea to a prototype
- Models for two PIC families available
- Can run stand-alone of connected to external hardware

PIC debugging interface



- Debugging needs facilities to send data to a terminal to be displayed, logged and evaluated
- Interface connects PIC circuit to the serial port of a PC
- Allows automated testing (in connection with ICSP)

Why an own development system?

- Some commercial development kits are available - but...
  - Are supported only on Microsoft Windows
  - Have no or very limited ICSP capabilities
  - Are slow (serial port only)
- Developing an own programmer costs **and** saves a lot of time
  - PIC does not need to be moved between devices
  - Very fast development cycles
  - Automated testing of subroutines is possible
  - 4500 lines of assembler code were written and tested for this project
- All pieces of hardware and software can be reused for other projects

Positioning a robot

Trilateration is a simple and effective way to detect positions

- 3 beacons are placed at different, known locations
  - The distance from every beacon to the robot is measured using the time of flight of an ultrasonic burst
  - Time of flight $\cdot$ sound velocity $=$ distance
  - Take a map of the area, draw a circle around each beacon with radius $=$ distance to the robot
  - All circles intersect in a single point, this must be the robot's location
- The receiver must know time when the burst leaves the sender, RF signals can be used for this purpose and to identify the sender

Roles of sender and receiver

Sender can be mounted on the robot or on the beacons

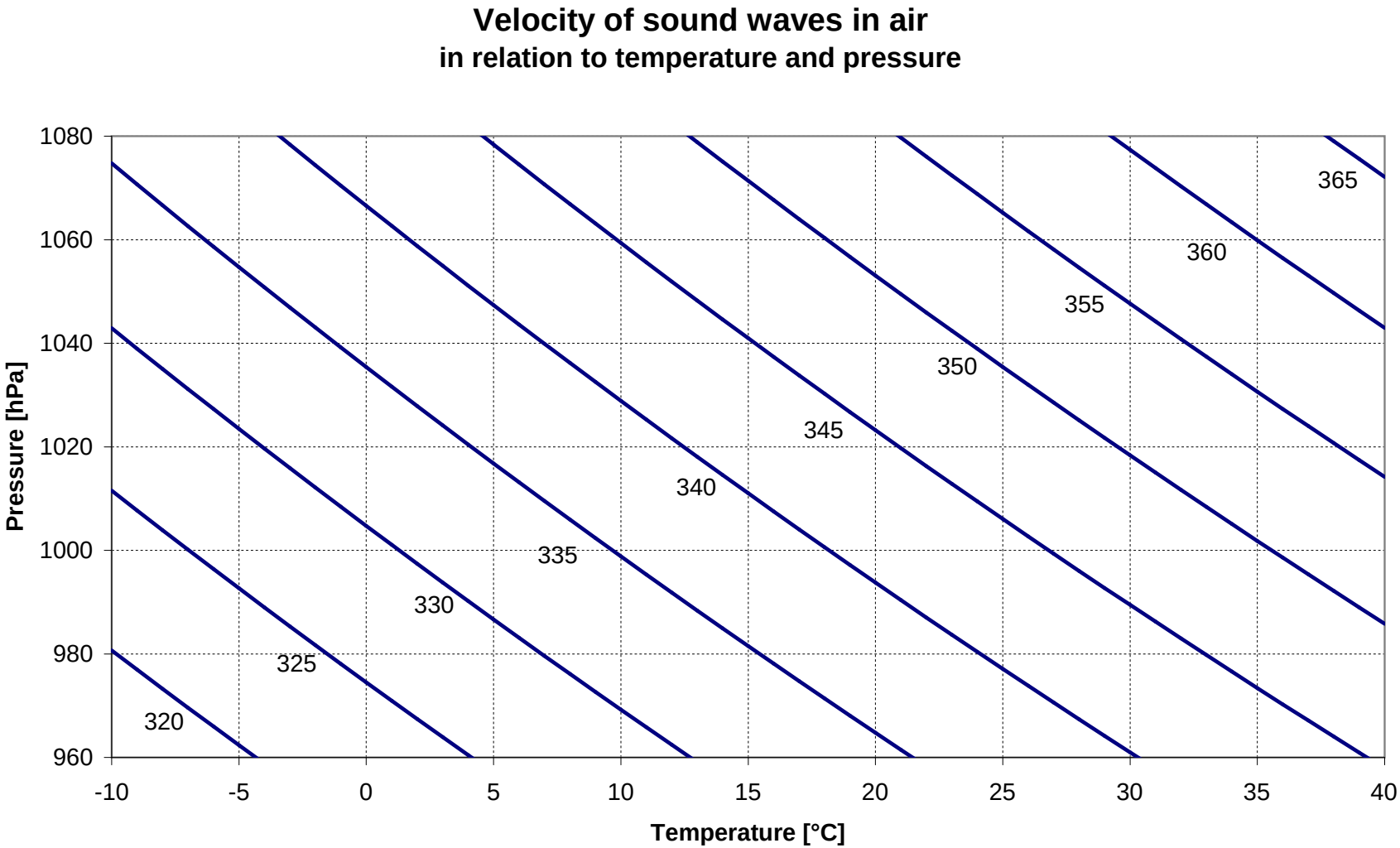
- Sender on beacon
  - ☹ Needs many impulses
  - ☺ Can locate many robots
  - ☺ RCX gets data directly
  - ☺ Simple hardware
- Sender on robot
  - ☺ Only one impulse
  - ☹ Only one robot
  - ☹ Data is sent to RCX
  - ☹ Difficult receiver

- Mobile senders and receivers were built, allowing a user to use and compare both alternatives
- Beacons need common control unit
- Receiver must be able distinguish which sender did actually send an ultrasonic burst

Measuring distances

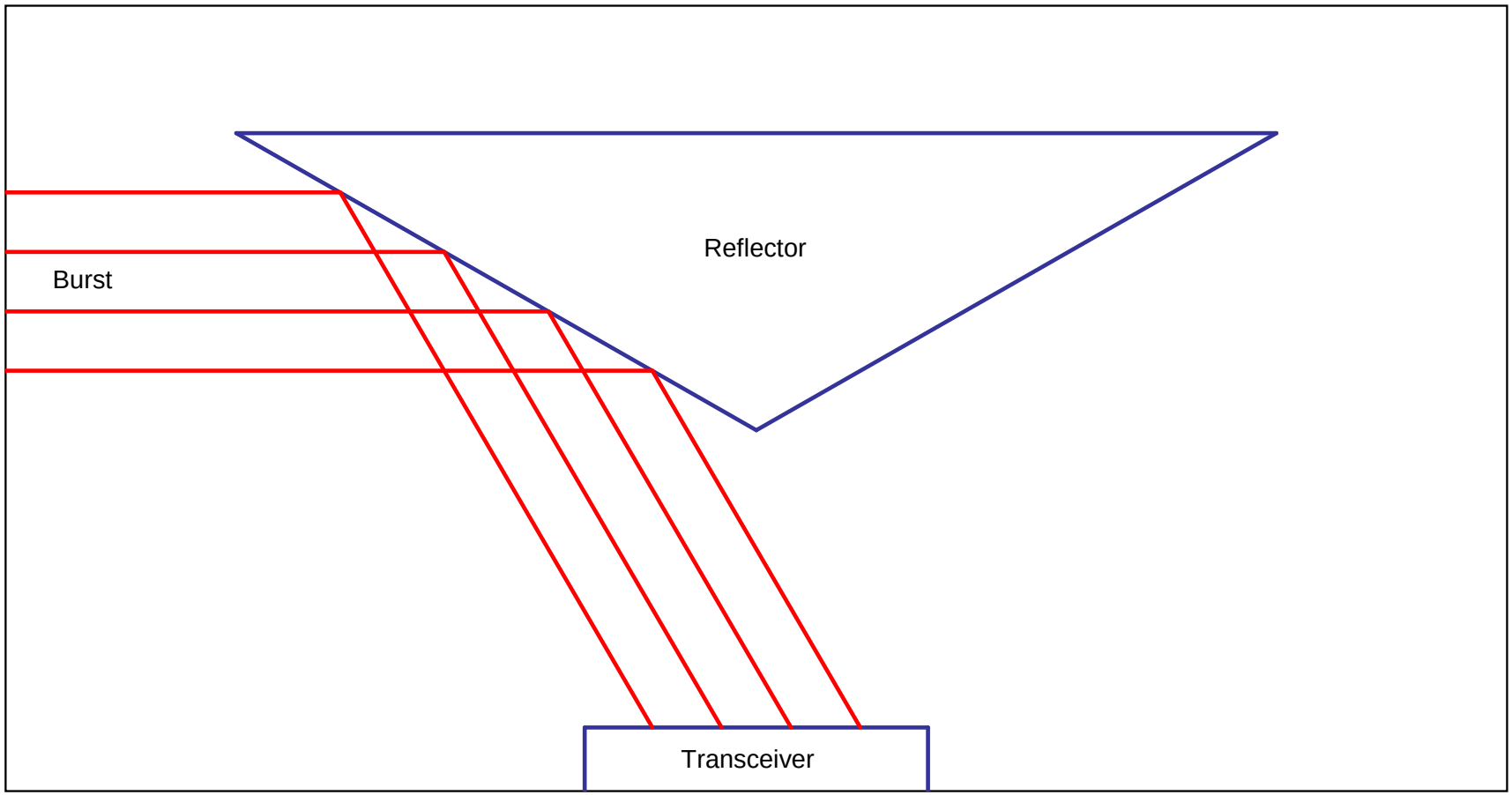
- Measuring distances is the base of positioning
- Ultrasonic bursts can be generated and detected by commercially available transceiver capsules
- A reflector cone is needed to circumvent directivity of sender and receiver capsules
- The sound velocity must be known (changes with temperature, pressure, density, ...), can be measured
- Maximum distance depends on sender strength and receiver sensitivity
- Minimum time between to bursts depends on echo duration

Sound velocity

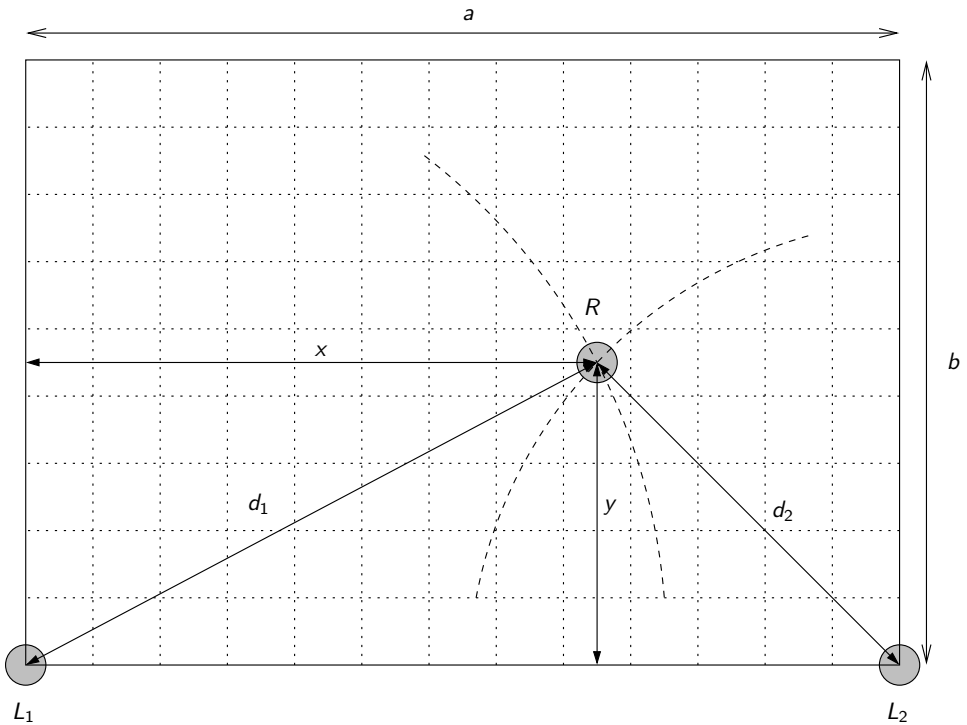


Reflector geometry

Simulation of a reflector cone
opening anle 120°, size 28mm, distance 14mm



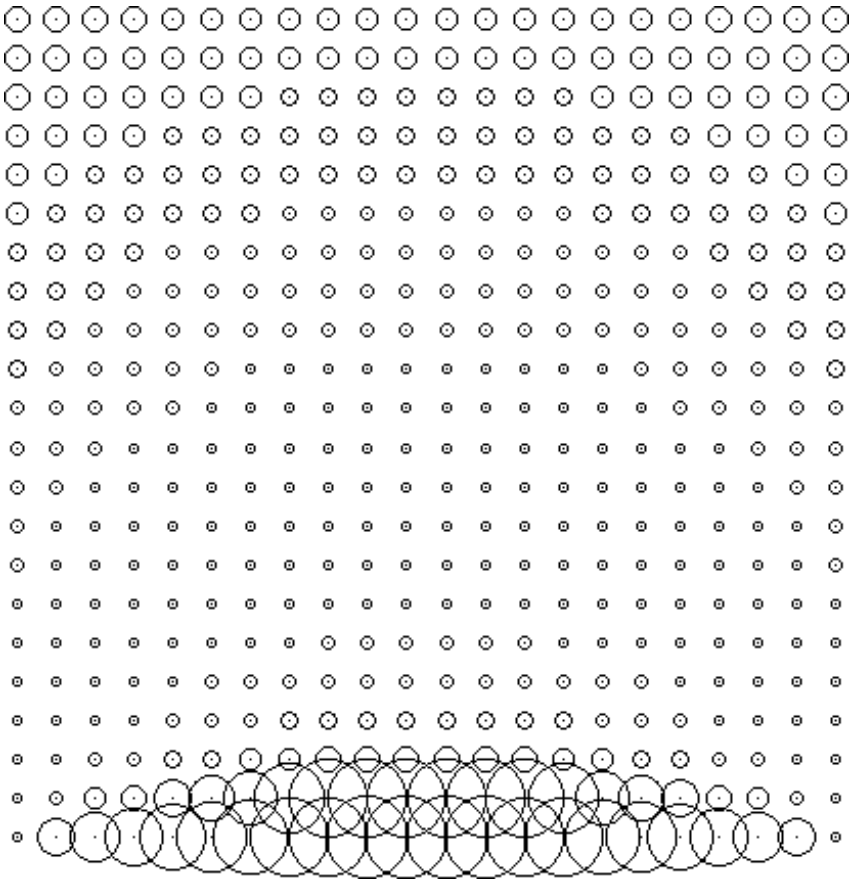
Calculating positions



- 2 beacons send ultrasonic and RF pulses
- $d_1 = c \cdot t_1$
- $d_2 = c \cdot t_2$
- Beacons identify themselves via RF

$$x = \frac{a^2 + c^2(t_1^2 - t_2^2)}{2a},\ y = \frac{\sqrt{4a^2c^2t_1^2 - \left(a^2 + c^2(t_1^2 - t_2^2)\right)^2}}{2a}$$

Accuracy



- The image shows errors in position, caused by 0.5% error in measured distances
- Huge errors between the two beacons

Improved scenarios

Additional beacons can improve the system

- $+1$ beacon in the top left or right corner can be used to avoid numerical problems
- $+1$ beacon makes it possible to measure the sound velocity
- $+1$ beacon eliminates the need to use RF signals

Ultrasonic positioning system

The system consists of

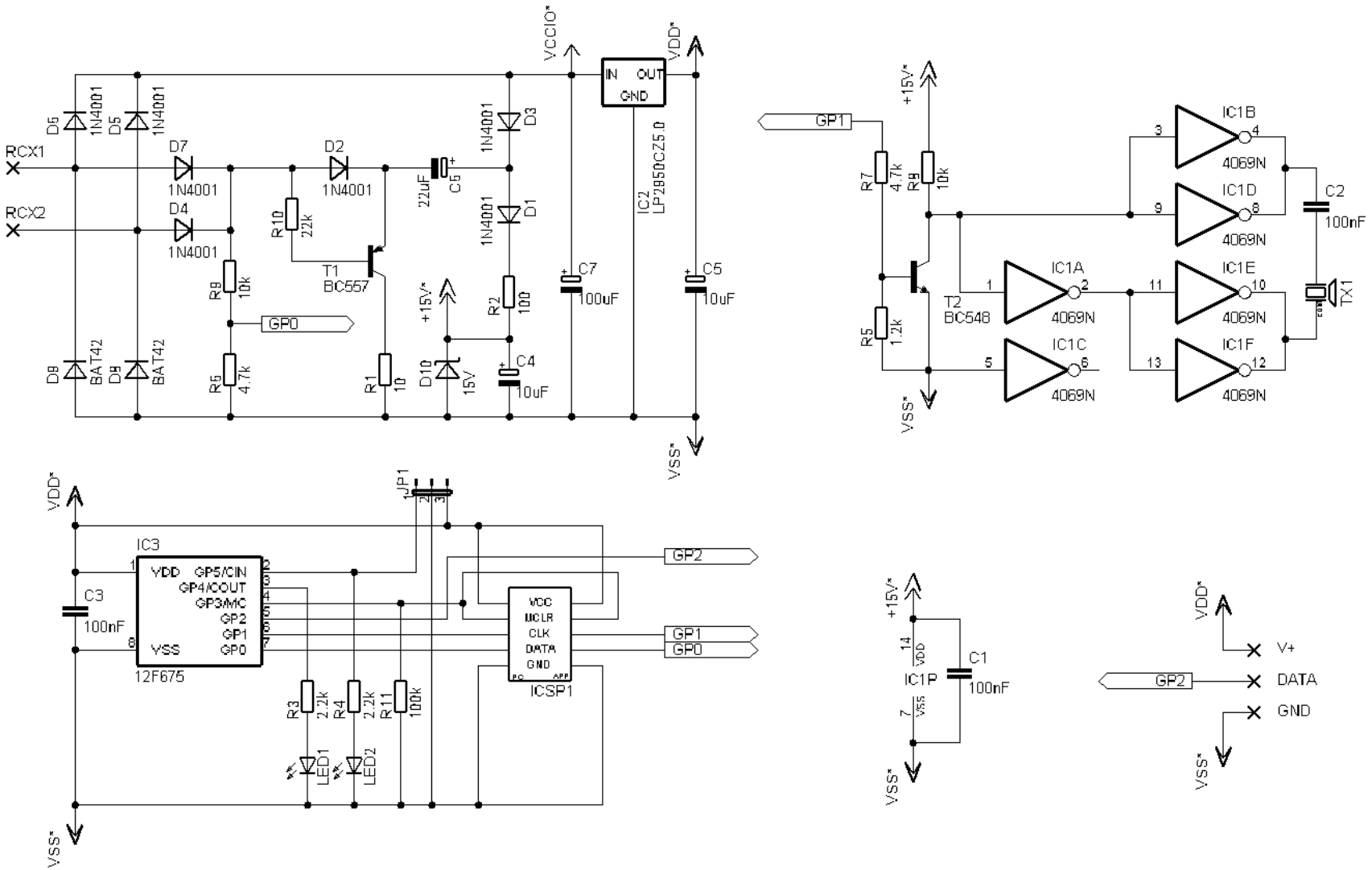
- 1 mobile sender (Mindstorms)
- 2 mobile receivers (Mindstorms)
- 4 beacons
- 1 beacon controller

It is possible to combine those devices to build many different scenarios.
Distances are measured by using ultrasonic bursts and synchronization information transitted via RF signals.

System description: Mobile sender (1/2)

- Sender is powered and controlled by the RCX by PWM
  - 4/8 PWM: charging capacitors
  - 7/8 PWM: activation signal
- Input voltage is doubled two times for US transmitter
- Prior to the ultrasonic burst an identification is sent using an 433 MHz RF transmitter
  - Serial data stream, Manchester coding
  - 1 start bit, 1 stop bit, 3 data bits with station ID
- Circuit is equipped with 2 LEDs, ICSP and debugger interface

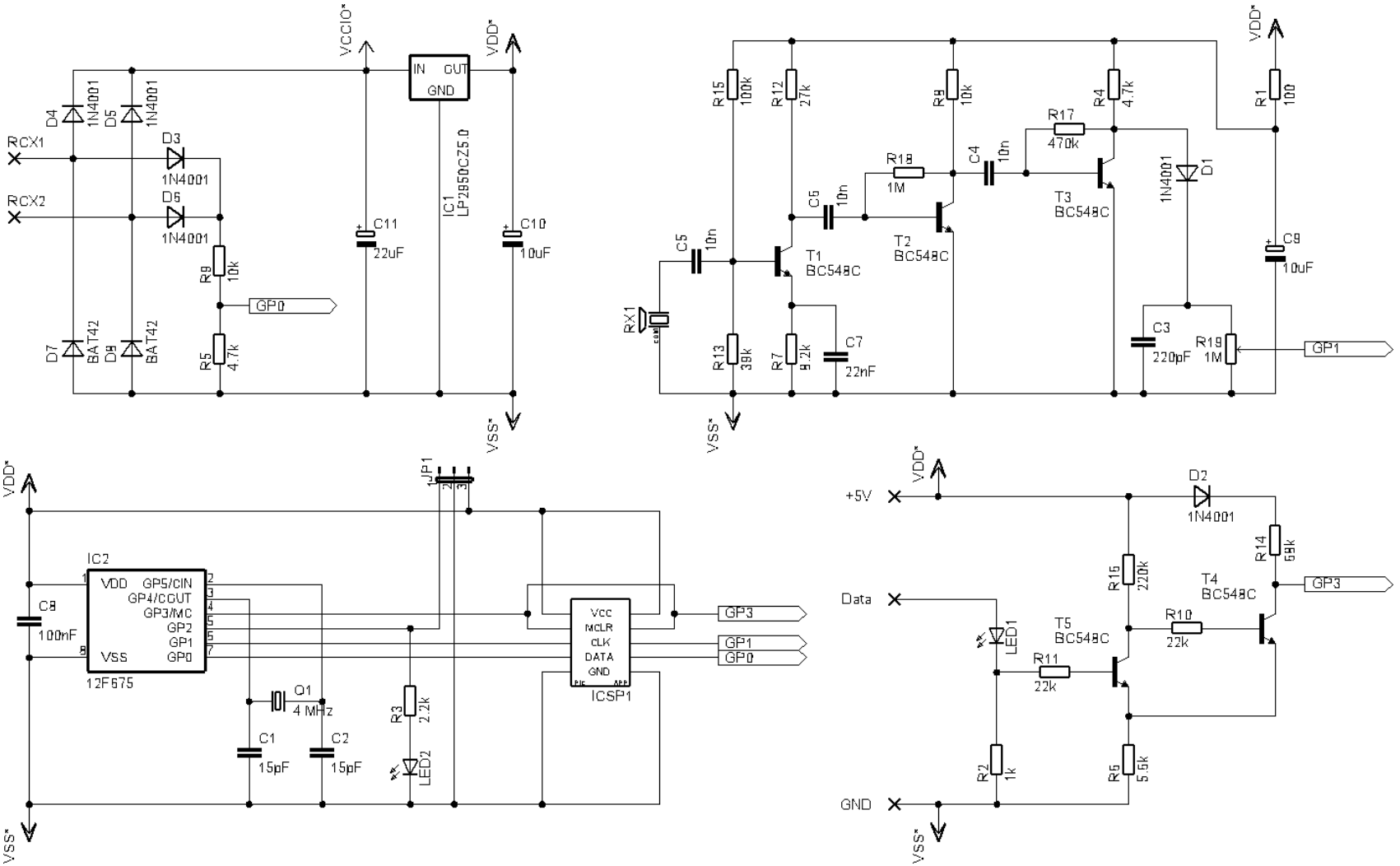
System description: Mobile sender (2/2)



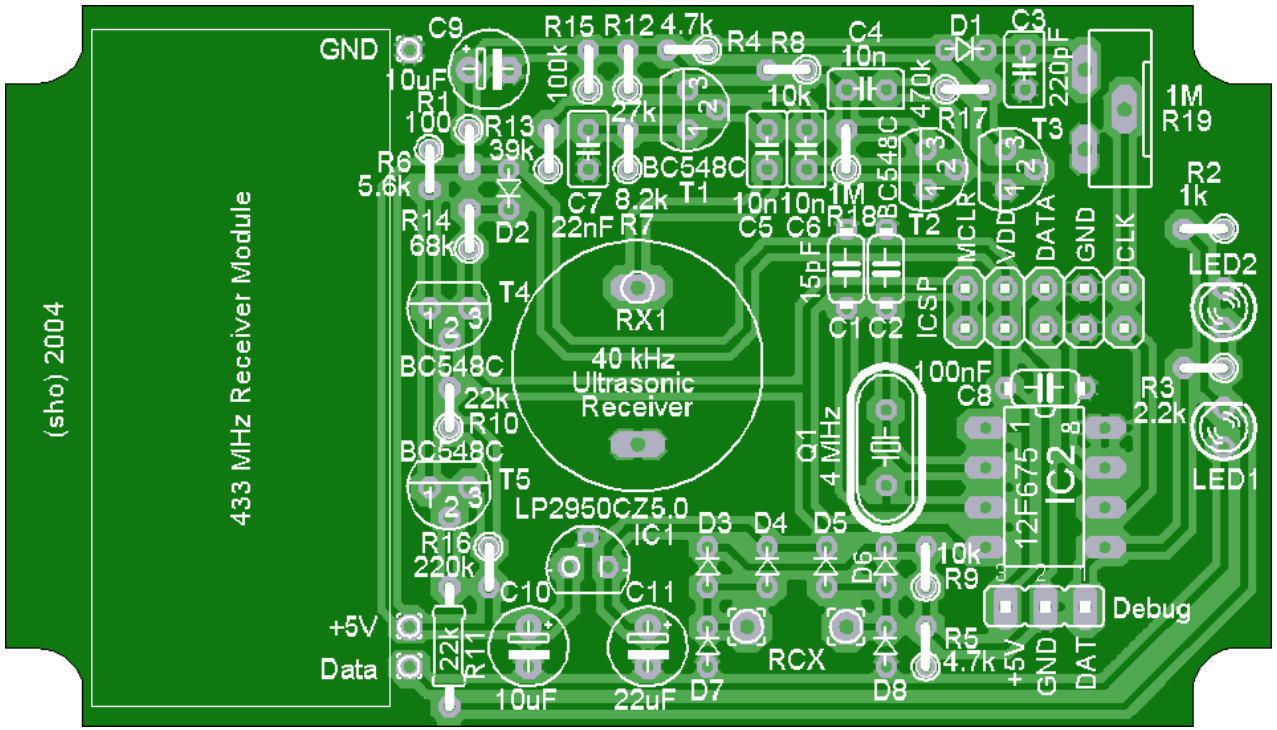
System description: Receiver (1/3)

- Receives ultrasonic burst and RF signal, measures times and sends information to the RCX (10 bytes packet)
  - Absolute time of flight of the sound burst in microseconds
  - Time between detections of the current and previous bursts
  - Duration of the echo in milliseconds
  - Status flags and identification of last sender
- Very low power consumption
- Highly optimized amplifier and detector
- Complicated software: synchronization with RCX (voltages and timing), measuring times with the needed resolution

System description: Receiver (2/3)



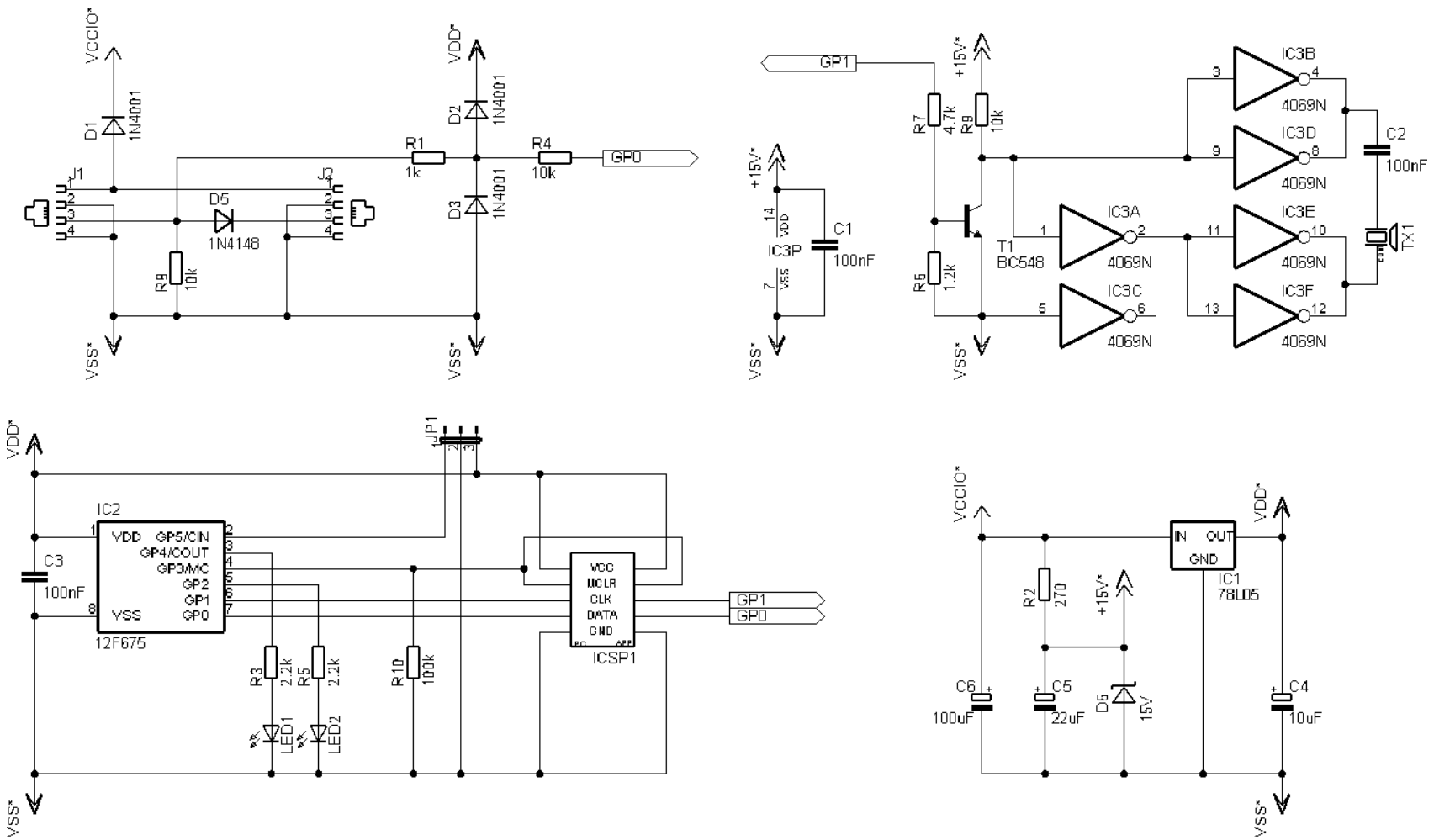
System description: Receiver (3/3)



System description: Beacon (1/2)

- Beacons are connected in a daisy chain, starting with the controller and ending with a terminator
- Beacons are numbered by their position in the chain
- The controller can send commands to fire a specific beacon
- The cable contains the power supply, the enumeration mechanism and a communication channel
- Beacons do only send ultrasonic bursts, RF transmissions are done by the controller

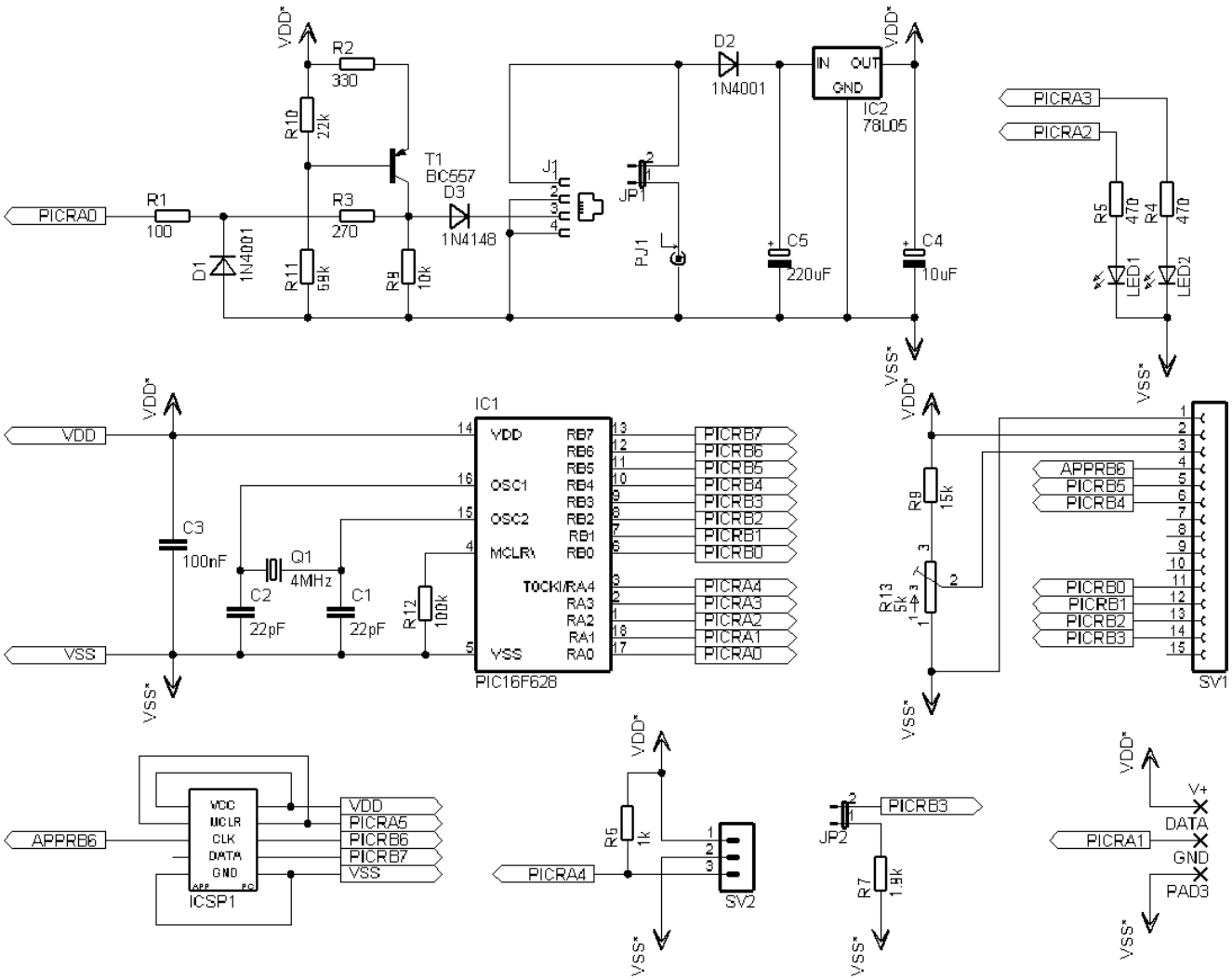
System description: Beacon (2/2)



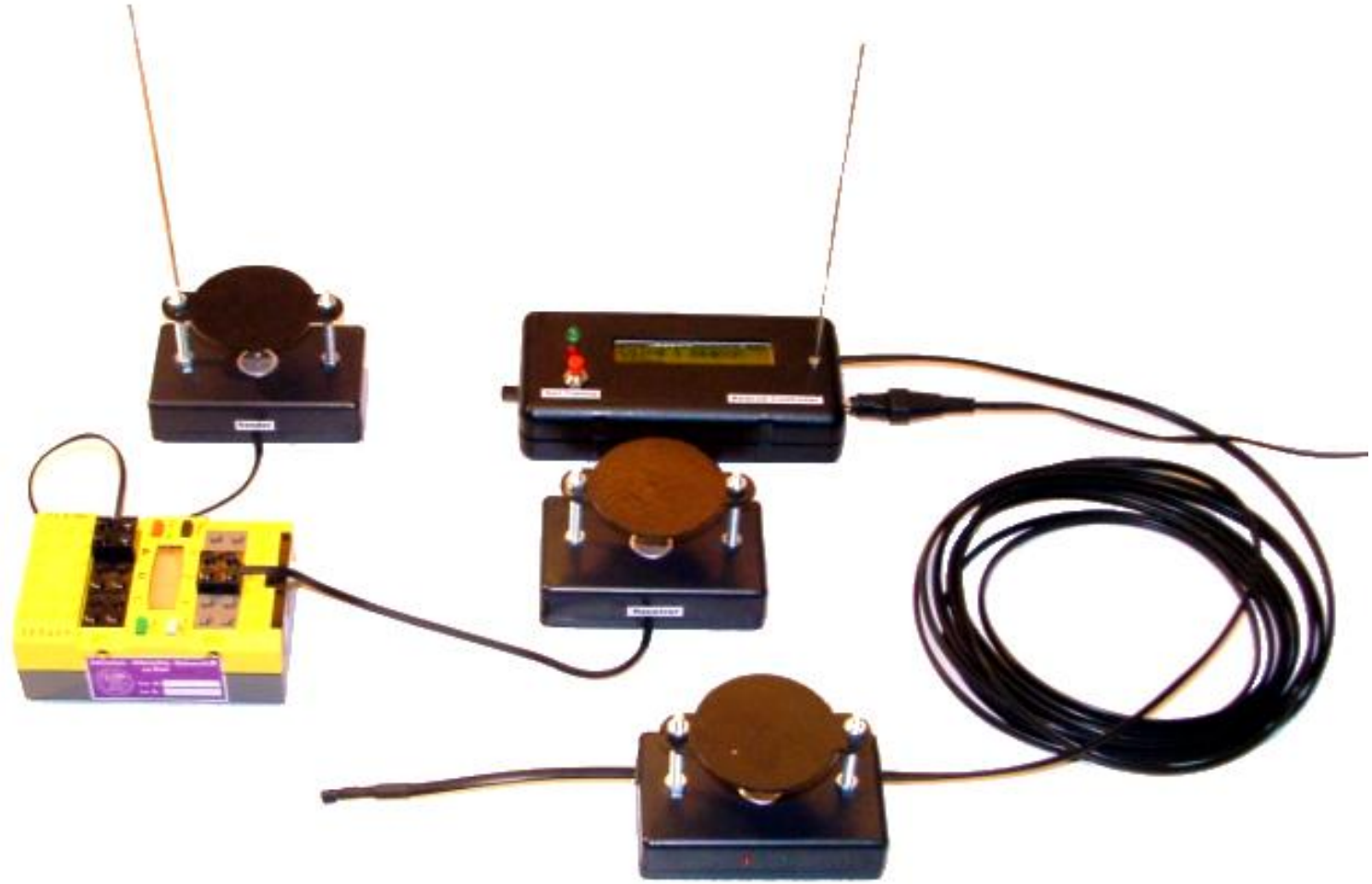
System description: Controller (1/2)

- Power supply for the beacons, controls system timing
- Contains RF sender
- Automated beacon counting and enumeration
  - 2 mA from a current source flow through the beacon chain
  - Every station decrements the voltage by 0.6 V with a diode
  - The current is grounded in the terminator
  - Every beacon can measure the voltage and calculate its number
  - The controller can calculate the number of beacons and change the current to send messages to the beacons
- A LC display shows status information
  - Wiring integrity status
  - Number of beacons
  - Active beacon firing schedule
- Schedule can be changed by a push button
- System configuration can be changed during runtime

System description: Controller (2/2)



Family picture



Performance and results

- Stable operating range is 10 m, mainly limited by RF range
- Distance measuring error is 1 mm in near range, slightly increasing with distance
- The whole system is flexible, many different scenarios are possible
- enough information is transmitted to the RCX to allow a variety of algorithms (e.g. positioning without RF signals, error detection,...)

Further work?

What remains to be done?

- Documentation is nearly complete
- The beacon controller needs a new case

Optional tasks

- Demonstration project with moving robot