Gforth EC auf dem NXT–Brick

Bernd Paysan

Forth–Tagung 2007

Vernichtende Kritik am RCX

Ullrich Hoffmann:

Wenn ich an den RCX–Wettbewerb in Haminkeln zurückdenke, fand ich das Forth und das Entwicklungssystem, das wir damals benutzten, primitiv und lästig zu bedienen. Für ein Wettbewerbs–Rapid–Prototyping war das nicht wirklich geeignet und als Vorzeige–Objekt, um andere für Forth zu interessieren, würde es wohl eher abschreckend wirken.

Vergleich NXT/RCX

RCX:

- Hitachi H8
- 16k ROM, 32k RAM (+512 Bytes schnelles RAM)
- Infrarot–UART
- PWM–gesteuerte Motoren ohne Drehgeber
- Robotics Invention System als Software

NXT:

- ARM7, ATmega48 als Coprozessor
- 256k Flash, 64k RAM
- Bluetooth SPP, USB
- PWM–geregelte Motoren mit Drehgeber
- Labview–basierte Software

Vergleich NXT/RCX

RCX:

- Hitachi H8
- 16k ROM, 32k RAM (+512 Bytes schnelles RAM)
- Infrarot–UART
- PWM–gesteuerte Motoren ohne Drehgeber
- Robotics Invention System als Software

NXT:

- ARM7, ATmega48 als Coprozessor
- 256k Flash, 64k RAM
- Bluetooth SPP, USB
- PWM–geregelte Motoren mit Drehgeber
- Labview–basierte Software

Vergleich NXT/RCX

RCX:

- Hitachi H8
- 16k ROM, 32k RAM (+512 Bytes schnelles RAM)
- Infrarot–UART
- PWM–gesteuerte Motoren ohne Drehgeber
- Robotics Invention System als Software

NXT:

- ARM7, ATmega48 als Coprozessor
- 256k Flash, 64k RAM
- Bluetooth SPP, USB
- PWM–geregelte Motoren mit Drehgeber
- Labview–basierte Software

Vergleich NXT/RCX

RCX:

- Hitachi H8
- 16k ROM, 32k RAM (+512 Bytes schnelles RAM)
- Infrarot–UART
- PWM–gesteuerte Motoren ohne Drehgeber
- Robotics Invention System als Software

NXT:

- ARM7, ATmega48 als Coprozessor
- 256k Flash, 64k RAM
- Bluetooth SPP, USB
- PWM–geregelte Motoren mit Drehgeber
- Labview–basierte Software

Vergleich NXT/RCX

RCX:

- Hitachi H8
- 16k ROM, 32k RAM (+512 Bytes schnelles RAM)
- Infrarot–UART
- PWM–gesteuerte Motoren ohne Drehgeber
- Robotics Invention System als Software

NXT:

- ARM7, ATmega48 als Coprozessor
- 256k Flash, 64k RAM
- Bluetooth SPP, USB
- PWM–geregelte Motoren mit Drehgeber
- Labview–basierte Software

Software–Umfeld

- Ausgereifte C–Compiler verfügbar (GCC)
- Lego–Firmware als „Open Source,“ allerdings eigene Lizenz
- Labview–Bytecode ebenfalls offengelegt (aber unbrauchbar).
- Andere Open–Source–Software wie LeJOS (JavaVM) unter MPL

Software–Umfeld

- Ausgereifte C–Compiler verfügbar (GCC)
- Lego–Firmware als „Open Source,“ allerdings eigene Lizenz
- Labview–Bytecode ebenfalls offengelegt (aber unbrauchbar).
- Andere Open–Source–Software wie LeJOS (JavaVM) unter MPL

Software–Umfeld

- Ausgereifte C–Compiler verfügbar (GCC)
- Lego–Firmware als „Open Source,“ allerdings eigene Lizenz
- Labview–Bytecode ebenfalls offengelegt (aber unbrauchbar).
- Andere Open–Source–Software wie LeJOS (JavaVM) unter MPL

Software–Umfeld

- Ausgereifte C–Compiler verfügbar (GCC)
- Lego–Firmware als „Open Source,“ allerdings eigene Lizenz
- Labview–Bytecode ebenfalls offengelegt (aber unbrauchbar).
- Andere Open–Source–Software wie LeJOS (JavaVM) unter MPL

Vergleich Gforth EC/hosted Gforth

Gforth EC:

- Primitives in Assembler
- Cross–Compiler erzeugt komplettes Image
- Alles selber machen $\longrightarrow$ Volle Kontrolle

hosted Gforth:

- Primitives in „prim,“ wird abgebildet auf C mit GCC–Erweiterungen
- GCC erzeugt Engine, Cross–Compiler erzeugt Image
- Fertige Libraries verwenden $\longrightarrow$ Arbeitserleichterung (?)

Vergleich Gforth EC/hosted Gforth

Gforth EC:

- Primitives in Assembler
- Cross–Compiler erzeugt komplettes Image
- Alles selber machen $\longrightarrow$ Volle Kontrolle

hosted Gforth:

- Primitives in „prim,“ wird abgebildet auf C mit GCC–Erweiterungen
- GCC erzeugt Engine, Cross–Compiler erzeugt Image
- Fertige Libraries verwenden $\longrightarrow$ Arbeitserleichterung (?)

Vergleich Gforth EC/hosted Gforth

Gforth EC:

- Primitives in Assembler
- Cross–Compiler erzeugt komplettes Image
- Alles selber machen $\longrightarrow$ Volle Kontrolle

hosted Gforth:

- Primitives in „prim,“ wird abgebildet auf C mit GCC–Erweiterungen
- GCC erzeugt Engine, Cross–Compiler erzeugt Image
- Fertige Libraries verwenden $\longrightarrow$ Arbeitserleichterung (?)

C–basiertes Gforth EC

Notwendige erste Schritte:

- Besorgen eines Cross–Compilers (gnuarm)
- Relokatibles Image muss in C–Code eingebunden werden: `fi2c.fs` anpassen
- Configure-Script anpassen, `machpc.fs` ebenfalls anpassen
- Terminal-Interface (USB oder Bluetooth)

C–basiertes Gforth EC

Notwendige erste Schritte:

- Besorgen eines Cross–Compilers (gnuarm)
- Relokatibles Image muss in C–Code eingebunden werden: `fi2c.fs` anpassen
- Configure-Script anpassen, `machpc.fs` ebenfalls anpassen
- Terminal-Interface (USB oder Bluetooth)

C–basiertes Gforth EC

Notwendige erste Schritte:

- Besorgen eines Cross–Compilers (gnuarm)
- Relokatibles Image muss in C–Code eingebunden werden: `fi2c.fs` anpassen
- Configure-Script anpassen, `machpc.fs` ebenfalls anpassen
- Terminal-Interface (USB oder Bluetooth)

C–basiertes Gforth EC

Notwendige erste Schritte:

- Besorgen eines Cross–Compilers (gnuarm)
- Relokatibles Image muss in C–Code eingebunden werden: `fi2c.fs` anpassen
- Configure-Script anpassen, `machpc.fs` ebenfalls anpassen
- Terminal-Interface (USB oder Bluetooth)

C–basiertes Gforth EC

Notwendige erste Schritte:

- Besorgen eines Cross–Compilers (gnuarm)
- Relokatibles Image muss in C–Code eingebunden werden: `fi2c.fs` anpassen
- Configure-Script anpassen, `machpc.fs` ebenfalls anpassen
- Terminal-Interface (USB oder Bluetooth)

Bluetooth

- Der NXT enthält einen „BlueCore“–Chip, der ein serielles Profil (SPP) implementiert
- Zwei Modi: Command und „transparent“
- Command–Modus: Message passing. Implementierung in der Lego-Firmware: State machine.
- Transparent: 1:1–Verbindung? Funktioniert noch nicht

Bluetooth

- Der NXT enthält einen „BlueCore“–Chip, der ein serielles Profil (SPP) implementiert
- Zwei Modi: Command und „transparent“
- Command–Modus: Message passing. Implementierung in der Lego-Firmware: State machine.
- Transparent: 1:1–Verbindung? Funktioniert noch nicht

Bluetooth

- Der NXT enthält einen „BlueCore“–Chip, der ein serielles Profil (SPP) implementiert
- Zwei Modi: Command und „transparent“
- Command–Modus: Message passing. Implementierung in der Lego-Firmware: State machine.
- Transparent: 1:1–Verbindung? Funktioniert noch nicht

Bluetooth

- Der NXT enthält einen „BlueCore“–Chip, der ein serielles Profil (SPP) implementiert
- Zwei Modi: Command und „transparent“
- Command–Modus: Message passing. Implementierung in der Lego-Firmware: State machine.
- Transparent: 1:1–Verbindung? Funktioniert noch nicht

USB

- USB–Controller im ARM enthalten
- Protokolldefinition „von Hand:“ kein brauchbarer Terminal–Standard vorhanden
- LibUSB liefert direkten Zugriff auf die USB–Schnittstelle

USB

- USB–Controller im ARM enthalten
- Protokolldefinition „von Hand:“ kein brauchbarer Terminal–Standard vorhanden
- LibUSB liefert direkten Zugriff auf die USB–Schnittstelle

USB

- USB–Controller im ARM enthalten
- Protokolldefinition „von Hand:“ kein brauchbarer Terminal–Standard vorhanden
- LibUSB liefert direkten Zugriff auf die USB–Schnittstelle

Die nächste Ebene

Was fehlt noch: Die eigentliche Roboter–Software

- Motorsteuerung
- Sensorauswertung
- usw. . .

Die nächste Ebene

Was fehlt noch: Die eigentliche Roboter–Software

- Motorsteuerung
- Sensorauswertung
- usw. . .

Die nächste Ebene

Was fehlt noch: Die eigentliche Roboter–Software

- Motorsteuerung
- Sensorauswertung
- usw. . .

Die nächste Ebene

Was fehlt noch: Die eigentliche Roboter–Software

- Motorsteuerung
- Sensorauswertung
- usw. . .