

LBS_{lip} als Open Source für Jedermann

LBS_{lip} an open Source to Everyone

Dipl.-Inform. **A. Spangler**, Dr. **M. Demmel**, Prof. Dr. **H. Auernhammer**,
Technische Universität München, Freising-Weihenstephan

Einleitung

Die Standardisierung der Elektronik in der Landwirtschaft gewinnt zunehmend für Traktoren- und Gerätehersteller an Bedeutung. Die Praxis fordert verstärkt die Bereitstellung entsprechender Steuerungen und Regelungen und macht Kaufentscheidungen von ihrer Verfügbarkeit abhängig. Diskussionen um DIN 9684 (LBS) oder ISO 11783 überdecken die wirkliche Problematik eines offenen Standards mit höchstmöglicher Kompatibilität durch einheitliche Interpretation der Norm und die Verwendung eines einheitlichen Codes.

Potential des offenen Kommunikationssystems „Landwirtschaftliches BUS System“ (LBS)

Das Kommunikationssystem LBS bietet durch seine Offenheit die Möglichkeit, Geräte unterschiedlicher Hersteller frei zu kombinieren. Zudem erlaubt es, neue Elemente für Zusatzfunktionen zu integrieren. Ein Task-Controller, der automatisch alle Arbeitsvorgänge dokumentiert, kann beispielsweise zur besseren Bewertung von Arbeitsmaßnahmen und für die „Gläserne Produktion“ verwendet werden. Intelligente Steuereinheiten, die mit Hilfe von Sensordaten (z.B. Chlorophyll- und Wassergehalt der Pflanzen) die Arbeitsweise von Anbaugeräten (z.B. Düngermenge) regeln, könnten mit Geräten aller Hersteller kombiniert werden.

Zudem können auch sog. „dumme Geräte“, welche keinen Jobrechner zur Steuerung benötigen, mit einem Implement Indicator (IMI) ausgestattet werden, um neben der Erfassung von Arbeitskenngrößen (z.B. Arbeitsbreite, -status, Transportstatus) auch Komfortfunktionen wie automatisierten Wechsel von Transport- in Arbeitsposition bei Gerätekombinationen ermöglichen. Durch geeignete Sensoren könnte dem Fahrer die aktuelle Maschineneinstellung und die daraus resultierende Arbeitsleistung vermittelt werden, was vor allem bei Leihmaschinen hilfreich ist, wenn keine detaillierten Kenntnisse über die Konstruktion der Maschine verfügbar sind, oder wenn von anderen Fahrern vorgenommene Änderungen an Maschineneinstellungen andernfalls nicht direkt erkennbar wären.

Sobald die Steuerung von Traktoren in der Lage ist, sich nach von Anbaugeräten vorgegebenen Bereichen für Arbeitsgrößen wie Zapfwellendrehzahl, Drehmoment und Fahrgeschwindigkeit zu richten, wäre mit LBS eine umfassende Optimierung von Arbeitsabläufen möglich. Wenn hierbei etwa eine Kreiselegge das maximal zulässige Drehmoment für die Zapfwelle vorgibt, kann ein Schaden vermieden werden, wobei die Regelung und die damit verbundene Sicherheitsüberwachung weiterhin lokal auf den Traktor beschränkt bleibt. Damit wird die mit einem offenen Regelkreis verbundene starke Abhängigkeit von Traktor und Anbaugerät vermieden, wodurch kein aufwendiges Versionsmanagement nötig wird. Da in diesem Fall die Kreiselegge keine weiteren lokalen Vorkehrungen gegen zu hohe Drehmomente treffen muss, können zudem Kosten gespart werden. Dieses Vorgehen bietet sich für viele Anbaugeräte an, die in gewissen Grenzen schwankende Eingangsgrößen tolerieren oder durch lokale Regelungen kompensieren können. Dazu müssen jedoch die beteiligten Geräte MIN- und

MAX- Sollwerte unterstützen und sollten einheitliche Strategien bei unerfüllbaren Vorgaben nutzen. Auf Basis dieser Technologie könnten Anbaugeräte mit Hilfe von Sensorinformationen selbst entscheiden, welche Eingangsgrößen es braucht (z.B. Zapfwellendrehzahl, Hubwerkstellung, Fahrgeschwindigkeit), um globalen oder lokalen Optimierungsvorgaben entsprechend die verlangten individuellen und teilflächenspezifischen Ausgangsgrößen (z.B. Bodenlockerung) zu liefern.

Probleme aktuell am Markt befindlicher Implementierungen von LBS

Schon bei der Entwicklung eines flexiblen Dokumentationssystems im Rahmen der Forschergruppe „IKB-Dürnast“, das dynamisch für die aktuell angeschlossenen Geräte passende Daten auf einer PCMCIA Speicherkarte aufzeichnet, wurden schwerwiegende Probleme erkannt (Tabelle 1):

Tabelle 1: Zusammenstellung der bei „IKB-Dürnast“ entdeckten Fehler von am Markt befindlichen LBS Geräten

Table 1: Compilation of incorrect LBS implementations which are detected during the research of IKB-Dürnast

Fehler	Auswirkung
Jobrechner („Electronic Control Unit“ –ECU) schickt Anfrage zu LBS Namen mit Absenderkennung des angefragten Teilnehmers (TN)	betroffener TN wertet dies als Adresskonflikt und weicht auf andere Kennung aus → dies kann zu endlosen Wiederholungen führen
Terminal agiert als TN mit Nummer 15 (reserviert für Broadcast) ohne sich vorher anzumelden	Jobrechner, die nur mit korrekt initialisierten TN kommunizieren, ignorieren diesen
Streckenzähler ([mm]) beginnt statt bei 32.767mm schon ab 10.000mm wieder mit 0	Anbaugeräte, die z.B. die Dosierung nach der Fahrstrecke richten, arbeiten falsch
Prozessgröße (z.B. Dosierung) wird ohne entsprechende Kennzeichnung als Gleitpunktzahl gesendet	eine Gleitpunktzahl, die durch fehlende Kennzeichnung als Ganzzahl ausgelesen wird, hat nichts mehr mit dem ursprünglichen Wert zu tun, so dass diese Information nicht verwertbar ist
Gerätetyp-Angabe im Identifizierungsfeld von Prozessgrößen entspricht nicht dem zugehörigen Gerätetyp	Botschaften zu dieser Prozessgröße werden nicht erkannt
unterschiedliche Kennungen der Größen Arbeits-/Gesamt –Zeit/Strecke	ein Dokumentationssystem muss gerätspezifisch entscheiden, wie diese Informationen erfasst werden können
Anbaugerät ignoriert auch korrekte Botschaften zu Prozessdaten von TN die nicht dem zugeordneten Terminal entsprechen	ein Dokumentationssystem kann passiv nur die Daten aufzeichnen, die jeweils am Terminal angezeigt werden; die Sollwerte eines intelligenten Steuerungssystems werden ignoriert
Anbaugerät verwendet keine Prozessdaten und wird über proprietäre Botschaften angesteuert	es ist weder eine Dokumentation der Arbeit noch eine Steuerung über Prozessgrößen möglich

Referenzimplementierung LBS_{lib}

Design-Ziele im Rahmen der Entwicklung eines Dokumentationssystems

In der Forschergruppe IKB-Dürnast sollte unter dem Stichwort „Gläserne Produktion“ ein automatisches Dokumentationssystem entwickelt werden, bei dem sowohl die Informationen von Applikationsgeräten über deren Jobrechner, als auch die Informationen von Geräten ohne eigene Steuerungs- und Regelungselektronik über einen (IMI) aufgezeichnet werden sollten. Hierbei muss eine große Anzahl von Informationen erfasst werden, wobei LBS-Basisdaten,

GPS-Informationen, gerätspezifische Prozessgrößen und nicht zuletzt unterschiedliche Sensorinformationen vom Traktor (z.B. Zugkraft) aufgezeichnet werden sollen. Für einen einfachen Umgang mit dem System sollten die Daten auf einer PCMCIA Memorycard gespeichert werden. All dies sollte bei jeder Arbeitsmaßnahme automatisch, auch ohne individuellen Auftrag, erfasst und aufgezeichnet werden.

Da keine den Anforderungen entsprechende Referenzimplementierung verfügbar war, wurde beschlossen, eine Programmbibliothek zu entwickeln, welche:

- den Standard möglichst umfassend umsetzt
- Teilaufgaben nur einmalig implementiert, um die Wartbarkeit der Software zu erhöhen
- flexibel einsetzbar ist:
 - Prozessdateninteraktion (Messwerte, Sollwerte) parallel mit beliebig vielen TN
 - Zusatzfunktionen:
 - Sollwertgeber benachrichtigen, wenn dieser nicht mehr eingehalten werden kann
 - explizit Sollwert aufheben (wenn keine Kontrolle mehr erforderlich ist)
 - explizite Information über die Ursache einer Sollwert-Ablehnung (nicht umsetzbar oder anderer TN hat Vorrang)
- auf Standard-Algorithmen zurückgreift, um Entwicklungszeit zu sparen und hohe Qualität zu erreichen
- über einen an der Aufgabenstruktur (z.B.: Botschaftsformatierung, LBS Systemverwaltung, LBS Prozessdaten) ausgerichteten modularen Aufbau verfügt, bei dem jede Teilaufgabe in einem möglichst kleinen Kontext ausgeführt werden kann, so dass nur ein kleiner Programmbereich zur Fehlersuche berücksichtigt werden muss. Diese am besten mit der objektorientierten Programmierung umzusetzende Technik vereinfacht eine auf mehrere Personen verteilte Administration, Erweiterung und Optimierung.

Aufbau der LBS_{lib}

Die in der LBS_{lib} realisierte Aufgabenverteilung wird in Abbildung 1 dargestellt, wobei die Elemente EEPROM_IO, Sensor_I, Actor_O, RS232_IO, LBS_Terminal optional sind.

Um eine möglichst einfache Anpassung auf unterschiedliche Rechner zu ermöglichen, und um den Umfang der identisch nutzbaren Quelltexte zu erhöhen, wurde eine Schichtstruktur entwickelt. Dabei wird in der untersten Anpassungsschicht mit wenigen Dateien eine einheitliche Schnittstelle zur Hardware geschaffen. Darauf aufbauend liefert eine Abstraktionsschicht wichtige Zusatzfunktionen. In der obersten Schicht, dem Großteil der LBS_{lib}, wird die Kommunikation implementiert. Dieser Teil kann überall unverändert eingesetzt werden, so dass Fehler schneller behoben werden und eine koordinierte Optimierung möglich ist.

Integration DIN 9684 entsprechender Elemente aus ISO11783

Durch den modularen Aufbau der LBS_{lib} war es möglich, im Januar 2001 den damaligen Status der Elemente von ISO 11783 zu integrieren, deren Funktion auch in DIN 9684 enthalten ist. Damit zudem ein gleichzeitiger Betrieb von DIN und ISO möglich wird, wurden Algorithmen entwickelt, durch die ein Gerät CAN optional mit 250kbaud initialisiert wird. Sobald ein DIN Gerät angeschlossen wird, das nur die vom DIN Standard festgelegten 125kbaud unterstützt, kann die LBS_{lib} die Baudrate im laufenden Betrieb reduzieren.

Die LBS_{lib} unterstützt die Anmeldung eines Systems und das Senden und Empfangen von Informationen, die den LBS Basisdaten entsprechen. Bei Prozessdaten kann die LBS_{lib} automatisch abhängig vom Empfänger entscheiden, ob eine Botschaft als DIN oder ISO Botschaft geschickt werden soll. Die Integration eines virtuellen Terminals ist ebenso möglich, wurde aber noch nicht durchgeführt.

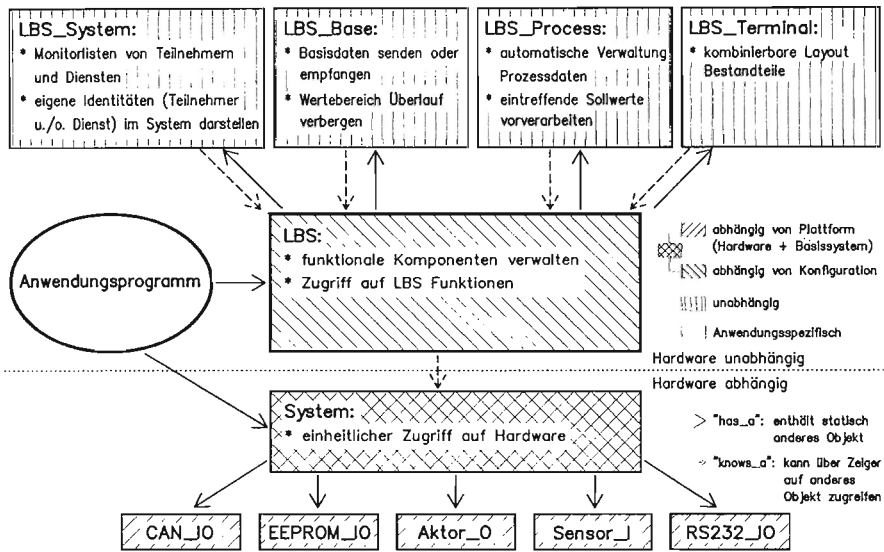


Abbildung 1: Funktionale Struktur der LBSlib
 Figure 1: Functional structure of the LBSlib

Integration von LBS+

Die Änderungen von LBS+ im Gegensatz zu DIN 9684 beziehen sich vor allem auf die Beschränkung auf gezielte Prozessdaten, die Integration eines GPS Positionsdienstes, das Klären von Details bei Prozessdaten und die Definition einer Alternative zum virtuellen Terminal, bei dem ein einheitlicher Upload von Bildschirmmasken ermöglicht wird. Die LBSlib ermöglicht bis auf den in der Planungsphase befindlichen Maskenupload die in LBS+ definierten Abwandlungen von DIN 9684.

Praxiseinsatz der LBSlib

Aufbauend auf der LBSlib und dem DIOS-IMI der Firma „Sensor Technik Wiedemann“ (STW) wurden IMI Systeme für verschiedene Gerätetypen entwickelt und mit Erfolg eingesetzt. Dabei können Informationen wie z.B. Ausbringungsmenge bei Gülleverteilsystemen oder die Arbeitsbreite bei Vario-Pflügen durch die Integration von Sensoren erfasst werden. Alle statischen, maschinenspezifischen Informationen (z.B. Gerätetyp, Name, Arbeitsbreite, Bedingungen für die Arbeitsstellung) können im EEPROM gespeichert werden. Da bei einigen Nachrüstlösungen für Traktoren die Hubwerkstellung nicht in den LBS Basisdaten enthalten ist, muss die IMI Software diese zur Ermittlung der Arbeitsposition vom Task-Controller erfragen, der diese Information über die Sensoren der EHR Regelung erhält.

Mit der LBSlib und der elektronischen Steuer- und Regeleinrichtung (ESX) von STW wurde ein Task-Controller realisiert. Die zu erfassenden Informationen werden zum einen abhängig vom jeweiligen Typ der Anbaugeräte nach Standardregeln ermittelt, und zum anderen durch flexibel im EEPROM gespeicherte Regeln anhand der Identifikation des Gerätes (Gerätetyp, Anbauposition und Name) ergänzt. Auf diese Weise wurden die auf der Versuchsstation aufgetretenen Inkompatibilitäten (s.o.) zum größten Teil ausgeglichen. Die Informationen werden mit GPS Position und Zeit verknüpft und anhand im EEPROM gespeicherter Eckpunkte von Rüstflächen und Feldern zu den Bereichen *Rüsten*, *Transport* und *Feld* zugeordnet. Auf

einer PCMCIA Speicherkarte werden die Daten sowohl sekundlich geloggt, als auch bei jedem Gebietswechsel als Zwischenergebnis festgehalten. Anhand der kompakt darin festgehaltenen Informationen kann man auch ohne eine Datenbank einen Überblick über die geleistete Arbeit erhalten.

Anforderungen der LBS_{lib} an Hardware, Entwicklungswerkzeuge und Programmierer

Da das BIOS der auf dem C167 (16bit CPU) aufbauenden Rechner von STW nur mit dem Compiler der Firma Tasking eingesetzt werden kann, konnten keine Erfahrungen mit dem Open Source Compiler gcc gemacht werden (von HighTec an die C16x Prozessoren angepasst).

Bei dem häufig eingesetzten *small* Speichermodell, bei dem zu jedem Zeitpunkt 64kByte effizient im **RAM** adressiert werden können, schränken die C/C++ Compiler Versionen von Tasking bis zur Version 7.0 den für C++ Objekte verfügbaren *HEAP* Bereich auf 16kByte ein, was für den Task-Controller knapp ausreichend ist. Die im Herbst 2001 veröffentlichte Version 7.5 hebt diese Beschränkung auch im *small* Modus auf, so dass genügend Platz für auf der LBS_{lib} aufbauende Applikationen verfügbar ist. Werden bei einem Großteil der Prozessdaten keine Sollwerte oder keine Messprogramme gewünscht, kann die LBS_{lib} so konfiguriert werden, dass der Bedarf an RAM erheblich reduziert werden kann. Bei den weniger effizienten *medium* und *large* Speichermodellen für eine 16bit CPU und bei einer 32bit CPU gibt es keine Beschränkung des *HEAP* auf 16kByte. Letztere bietet sich zudem an, da viele Daten in LBS durch 32bit Zahlen dargestellt werden, so dass z.B. zeitabhängige Aktivitäten und Prozessdatenwerte effizienter bearbeitet werden können.

Solange die LBS_{lib} auch Elemente von DIN 9684 unterstützt, die in der Praxis nie implementiert werden (z.B. System Kommandos zu Status und Stop, spezielle Alive Botschaft für TN im Fehlerzustand), belegt allein die LBS_{lib} im Flash **ROM** ca. 106kByte (siehe Abbildung 2).

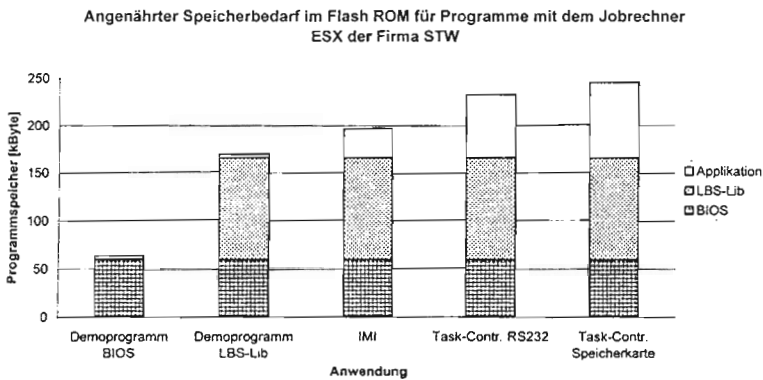


Abbildung 2: Flash ROM Programmspeicher unterschiedlicher Programme

Figure 2: Flash ROM program memory for different programs

Werden Teilfunktionen in einer zentralen Konfigurationsdatei deaktiviert, kann die Größe der LBS_{lib} sehr stark reduziert werden.

Open Source zur Förderung von LBS und der LBS_{lib}

Open Source kann auch unabhängig von der LBS_{lib} wichtig für eine Weiterentwicklung von DIN 9684 und ISO 11783 sein, wenn analog zu Apache, LINUX oder TCP/IP offene Diskussionsforen im Internet eingerichtet werden, in denen Interpretationsspielräume geklärt und

Ergänzungen zeitnah auch von Software Entwicklern diskutiert werden können. Angelehnt an TCP/IP sollten durch „Request For Comments“ (RFC) Änderungen erst mit einer Referenzimplementierung und nach Diskussion und Akzeptanz des Vorschlags in den Standard eingebracht werden können. Auf diese Weise würden Probleme, die erst bei der Implementierung des Protokolls erkennbar würden, schon von vornherein erkannt.

Mit der LBS_{lib} ist eine umfassende Implementierung von DIN 9684 verfügbar, die es bei einem breiten Einsatz als Referenzimplementierung allen Entwicklern ermöglicht, sich mehr auf die spezifischen Funktionen einer Maschine zu konzentrieren. Die dazu erforderliche Qualität kann am besten von einer Software Firma gewährleistet werden. Dies könnte analog zuden Lizenzmodellen von Programmbibliotheken für J 1939 erfolgen. Bei der LBS_{lib} könnten Kunden professionellen Support mit Schulung, Projektbegleitung und zugesicherten Qualitätsstandards (z.B. durch Zertifizierung von LBS_{lib} Versionen) gegen Gebühr erhalten. Durch die „Lesser General Public“ Lizenz (LGPL) der LBS_{lib} müssen alle Optimierungen der LBS_{lib} des Kunden evtl. gegen Erhalt einer Prämie offen zugänglich gemacht werden, so dass eine einheitliche und schnelle Verbesserung der Software unterstützt wird.

Ausblick

Die LBS_{lib} hat sich im Praxiseinsatz im IKB-Dürmast auf der Versuchsstation Dürmast bewährt. Einem Einsatz der LBS_{lib} in der Industrie steht nichts mehr im Wege stünde. Unabhängigbar ist nach unserer Meinung eine Software Firma, welche einen professionellen Support anbieten kann.

Literatur

Auernhammer, H., Demmel, M., Spangler, A.: Betriebsdatendokumentation mit LBS und GPS für Traktor-Gerätekombinationen. Tagung Landtechnik 1999. In: Düsseldorf: VDI-Verlag 1999, S. 217-221

Auernhammer, H., Demmel, M., Spangler, A., Trukenbrod, R.: Automatic process data acquisition with GPS and LBS. Abstracts of AgEng Warwick 2000: Agricultural Engineering into the Third Millenium. Silsoe: Silsoe Research Institute 2000, Part 1, pp.267-268

Spangler, H., Auernhammer, H.: Die LBS-Programmbibliothek - erstes Open Source Modell in der Landwirtschaft. In: Berichte der Gesellschaft für Informatik in Land-, Forst- und Ernährungswirtschaft. Referate der 21. GIL-Jahrestagung in Freising-Weihenstephan, Weihenstephan 2000, Band 13, S. 210-214

Spangler, A., Auernhammer, H.: Die LBS Programmbibliothek - erstes Open Source Modell in der Landwirtschaft. In: Proceedings of the 2nd Sapporo International Symposium 2000: Electronic Farm Communication with LBS. Hokkaido (Japan): Hokkaido Branch of JSAM 2000, S. 89-115

Spangler, A., Auernhammer, H., Demmel, M.: Program Library “LBS-Lib” for Agricultural BUS System (LBS, DIN9684) – first Open Source Project in Agriculture. Proceedings of the Third European Conference on Precision Agriculture, Montpellier 2001

Summary:

The free available Open Source program library LBS_{lib} was successfully used for an comprehensive automatic documentation system at the agricultural experimental station Dürmast. The LBS_{lib} has the potential to serve as a reference implementation of DIN 9684 and its counterparts in ISO 11783, if professional support is provided by a software company. This enables a smooth transition from already existing DIN 9684 systems to future ISO 11783 devices. With the possibility of excluding unused parts of the LBS_{lib}, it can be also used in combination with a 16bit CPU.