MEASUREMENT OF THE PHOTOMETRIC DISTRIBUTION OF LIGHT SOURCES BY DEFLECTOMETRIC TECHNIQUES

Arranz, M. J.,Arasa,J., Royo,S. and Caum, J.
Center for Sensors, Instrumentation and Systems Development (CD6)
Universitat Politècnica de Catalunya, c/ Sant Nebridi 10 E-08222 Terrassa (SPAIN)
T.+3493 7398355 F.+3493 7398923 E-mail: {arasa, royo,caum,mjesus}@oo.upc.es

Abstract **–This paper proposes an instrument allowing the accurate characterization of the energetic distribution of a given source at a distant plane positioned at will. This is achieved by measuring the light pattern leaving the source by applying deflectometric techniques and photometric measurements, and then applying raytracing algorithms in order to propagate this distribution to a distant plane. The techniques allow a very compact setup, and a prototype of the optomechanical system has been implemented and tested. The system is intended to substitute the photometric tunnels now in use for testing, for instance, automotive reflectors.**

1.INTRODUCTION

Present-day extended source testing intrumentation usually are long-distance (nominally, 10m) tunnels where an extended source is placed at one edge of the tunnel and the resulting photometric response is measured at the opposite wall. Manufacturers using extended sources (from lamp or LED reflectors to automotive lighting) need carrying such instruments to a distant setup. The problem may be relevant in, for instance, prototype testing, where the prototype needs to be carried to a distant installation in order to evaluate its response.

From the work developed by our group in the past years, both on deflectometric measurements of optical quality surfaces [1] and photometric simulations on extended sources [2] , the possibility of building a goniometric instrument with the capability of measuring simultaneously both the outcoming wavefront from the source and its photometric distribution arose. Such an instrument may take advantage of raytracing and energetic flux propagation techniques [3].

In the following Sections, the measurement procedure will be described in a Measuring Principle section. Next, the typical measuring arrangement will be presented in an Experimental Setup section. A short introduction to the data processing operations will be given in an specific section, prior to presenting the experimental results obtained up to date in the prototype setup for a 10W halogen lamp energetic distribution at a plane distant 10m. Finally, the future works in the system will be presented and the conclusions from the work at this stage.

2.MEASURING PRINCIPLE

As a combination of deflectometric, photometric and raytracing techniques is the basis of the measurement, the basis of each technique will be introduced in an independent manner.

A. Deflectometric Techniques

Deflectometric techniques are based on the simultaneous measurement of the position and slope of a bundle of rays. Typical applications involve topographic profiles of wavefronts emerging from optical systems, from which surface profiles may be extracted, or the involved optical system characterised. So, the result of the measurement will be the determination of the slope (u,v) of a set of rays at a given point (x,y), where the sampling of the wavefront has been performed. This means the result of the deflectometric measurement is a set of four values for each of the rays involved.

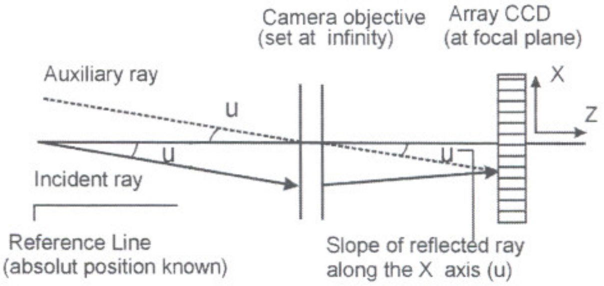


Fig.1 Deflectometric measurement of slope and position

The slope measurement is achieved by registering the light pattern behind a diaphragm placed at a known position which fixes the (x,y) data using an objective focused at infinity which allows performing slope measurements on the CCD array placed at the focal plane of the objective (fig.1).

This means the *u* term (the slope in the X axis direction), will be given by an expression as simple as

$$u = n_x \frac{\Delta x}{f'} \quad (1)$$

0-7803-7241-7/01/$10.00 (c)2001 IEEE

where n_x is the pixel number from the center of the array, Δx is the pixel size along the X axis and f' the effective focal length of the objective. Measurements of ν would be equivalent along Y axis.

An important issue to be adressed is an adequate sampling of the emission of the source. This means positioning the camera in enough positions of the emitted field of the camera so as to obtain a smooth representation of the emission.

B. Inclusion of energetic features

The natural way to include the energetic distribution in the measured wavefront was the measurement of the energetic contribution of each of the rays measured by the deflectometric technique. This introduces a fifth "coordinate" for each ray describing the luminous flux it carries, yielding a (x,y,u,v,F) description for the bundle of rays. For our prototype, the flux value is extracted from the gray value present in the pixel in the CCD array where the ray slope is measured.

Obviously, at this point relevant assumptions regarding energetic measurements need be performed in the prototype. A rigorous calibration procedure of the CCD and the A/D conversors (in the camera and in the frame grabber) need be performed in order to obtain absolute flux measurements from the gray-level digitized signal. Additional problems, such as the calibration unstability in time and the limited range of the detector need be studied and addressed.

Obviously, the smoothness in the photometric distribution of the emitted field need be taken into account when deciding the sampling density, in order to consider all the relevant effects in the source (filament shape, for instance). In aditon, the digitization level of the source needs be considered as the amount of truncated information may be important if the level of digitization used is too low.

C. Propagation

So, the application may characterize adequately the emitted light distribution of the source at a set of positions, by setting up a five data set for a selected bundle of rays. The final step is ray and energy propagation from the measured plane to a final surface where the photometric distribution needs being validated.
The energy distribution on the final surface is obtained by propagation of all the registered rays to the desired surface using conventional raytracing techniques, available as the slope (u,v) and position (x,y) information of each point have been measured. Flux is also propagated along the rays and distributed on the final surface in order to calculate the final photometric distribution.

Again, the sampling requirements of the system imply that the number of rays need to be high enough in order to obtain an adequate energetic distribution, as each measured ray turns into an impact on the final surface.

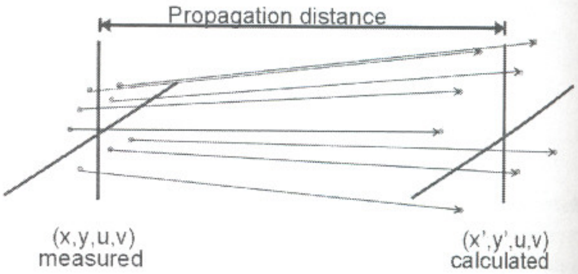
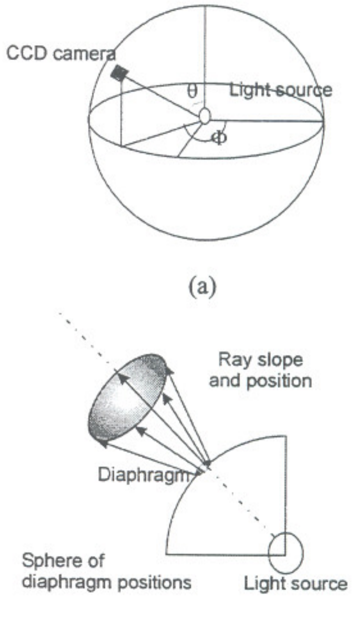


Fig.2 Propagation of rays from the measured surface to the final surface

3.EXPERIMENTAL SETUP

The design selected for the experimental setup of such an instrument is a compact motorized goniometer. This will enable the combination of deflectometric and energetic measurements required with the ease of adjusting the sampling density at will by positioning the detector in a group of points in the emitted field.



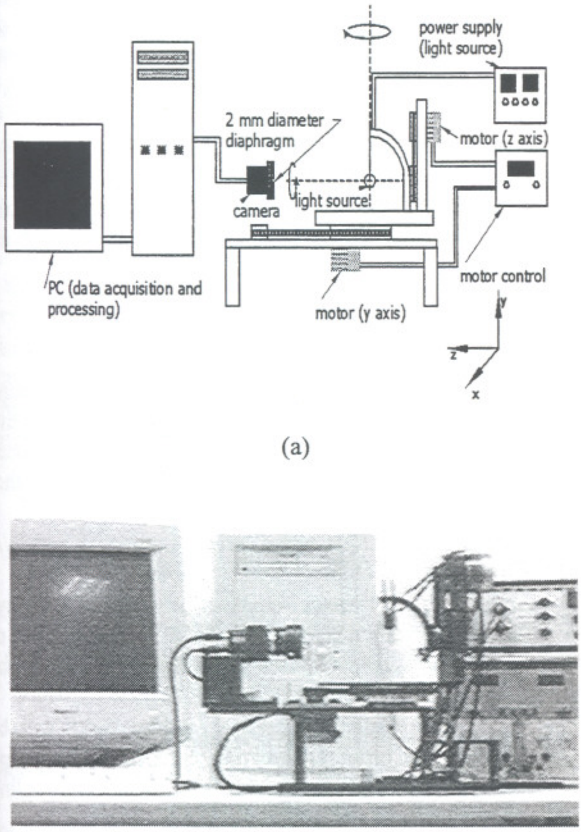
(a)

(b)

Fig.3 a) Goniometer; b) Detail presenting the diaphragm limiting the slope range

The number of slopes acquired at each position is limited by a 2mm diameter diaphragm placed in front of the objective, combined an appropriate selection of the source to diaphragm distance. It should be considered that the deflectometric mesurement technique presented in eq.1 is intrinsecally limited by the number of pixels in the CCD array. The distance from the source to the diaphragm and the diameter of the diaphragm are are the parameters to be adjusted in order to set the proper sampling of the emitted field, both in the number of acquisitions and in the control of the available slope range.

Due to obvious mechanical stability reasons, it was prefered to move the source while keeping the CCD stable. This allowed easy and precise positioning of the diaphragm in a group of positions in the emitted field of test (fig.4).



(a)

(b)

Fig.4 a)Setup layout; b)Experimental setup

The used setup is depicted schematically in figure 4a and will be described at length below. As a general overview of the setup, an intensity source power supply feeds the source under test, in our case a 10W halogen lamp with a filament 6mm long. The source has been placed 114.6mm away from the objective. A Presentco CCD camera with a standard 16mm objective and a tailored circular diaphragm (diameter 2 mm) registers the light patterns from which slope and flux information from the source are extracted.

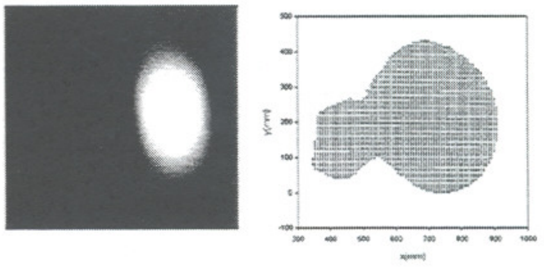
In order to allow easy and accurate data acquisition, movements of the source around Z axis and Y axis have been motorized, using two stepping motors allowing angular increments of 1°. This allows automating the measurement so high sampling densities may be attained minimizing error in the acquisition.

The present instrument design covers 120° of the angles around Y axis and around Z azis, centered at the axis of symmetry of the source. A set of 13 angular positions around Y axis and 13 angular positions around Z axis (10° angular increments around each axis) have been acquired, yielding 169 sampling positions in the emitted field. The registered information is recorded and processed using an IP8 frame-grabber digitizing at 8-bit depth yielding a 512x512 pixels image. Each of the rays from the image provides a set of (x,y,u,v,F) values measured as described in Section 2, which will be propagated to any desired final surface.

In the presented experiment, a surface equivalent to that used in photometric tests of automotive lighting has been used, that is, the measured energy has been propagated to a plane distant 10m. However, propagation might be performed to arbitrary surface shapes and distances.

4.EXPERIMENTAL RESULTS

From the described setup the characterization of the distribution of the lamp in a distant system has been attained. A typical individual register may be observed in fig.5a, and the propagation of each of the rays to the final surface is presented in fig.5b.



a) b)

Fig.5 a)Typical single register, b)Propagated ray pattern at a plane 10m away.

As far as the information contained in the rays obtained from a given register becomes quite similar, an array of 5x5 rays has been selected in each image in order to reduce the amount of data an enable quick data processing. The total set of (x,y,u,v,F) values from the selected rays in each of the 169 images may then be measured and the rays propagated to the final plane in the same way as the ones presented in fig.5.

The final results for the propagated data are presented as a normalized illumination pattern both in the format of a three-dimensional representation (fig.6) and as a pseudocolor pattern where the results have been encoded as an 8-bit grayscale in order to yield a clearer image interpretation (fig.7).

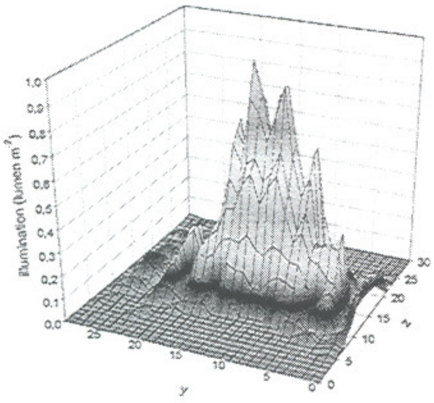


Fig.6 Normalized illumination in the final plane:3D view

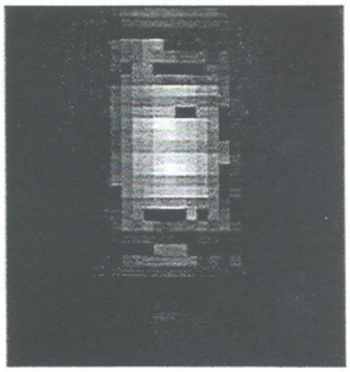


Fig.7 Normalized illumination in the final plane: representation as an 8-bit grayscale

Obviously, fig,6 and fig.,7 are non-smooth figures, yielding some abrupt changes in the final photometric distribution which would not be observed in the present system. This is a consequence of the low degree of sampling used in the experiment, both in the positioning of the diaphragm (there are 10° increments between neighbouring images) and in the ray density obtained from each image. However, the validity of the measuring principle becomes demonstrated, as, except for sampling defects which are easy to correct by simply rising the number of data acquisition steps, the photometric distribution on the final plane has been reconstructed on a distant plane from simple measurements performed on a goniometer. A low-cost and compact alternative tool for light source testing without the presence of photometric tunnels has demonstrated its feasibility.

5. CONCLUSIONS

A new method for light source evaluation without the need of large installations, such as photometric tunnels, has been proposed, and a test prototype allowing the validation of the feasibility of the technique implemented.

The method is based on the measurement of a set of rays outcoming from the light source under test. Measurements are a combination of deflectometric techniques, where the ray slope at given positions of rays emitted by the source are obtained, and photometric techniques, where the amount of energy present in each ray is obtained. This yields a (x,y,u,v,F) data set for each ray, that allows a propagation of the energy to the surface of interest.

Qualitative energetic distributions on a plane distant 10m (simulating the photometric tunnels used in automotive reflector testing) have been obtained, showing the importance of the optimization of the sampling of the system.

6. FUTURE WORK

As a presentation of a work in progress, there are many important possibilities to follow on and turn the qualitative results obtained up to date into quantitative values, once the measuring technique has been tested. Among other possibilites, our future work will

- Calibrate the global setup, mainly the A/D and D/A conversions of the involved electronics.
- An optimized detector (such an on-chip CCD array) will be implemented in the system. A calibration of the gray level response to a given flux stimulus will be performed.
- The sampling density of the emitted field will be optimized in order to enhance the final photometric response. Theoretical studies on optimum sampling densities will be performed.
- A verification tool of the quantitative results obtained will be developed in order to validate the final surface energetic distributions.

7. BIBLIOGRAPHY

[1] J.Arasa, S.Royo and N. Tomàs, "Simple method for improving the sampling in profile measurements using the Ronchi Test", Appl. Opt. vol. 39. pp. 4529-4534.

[2] J.Arasa, S.Royo, C. Pizarro and J. Martínez, "Flux spatial emission from technical specifications for a general filament light source", App. Opt. 38, 7009-7017.

[3] D.Malacara, Optical Shop Testing. New York, Wiley, 1992.

[4] W.L.Wolfe, Introduction to Radiometry , vol TT29, Tutorial Test in Optical Engieering, Washington, SPIE, 1998.

Integration of Java processor core JSM into SmartDev(ices)

Frank Golatowski, Stephan Preuss*, Hagen Ploog, Thomas Geithner*,
Clemens Cap*, Dirk Timmermann

Dept. of Electrical Engineering & Information Technology
University of Rostock
R.-Wagner-Str. 31, 18119 Rostock-Warnemuende
Germany

* Dept. of Computer Science
University of Rostock
A.-Einstein-Str. 21, 18059 Rostock
Germany

Abstract – **This article describes the current work to extend the Java processor Java Silicon Machine (JSM) for usage in embedded systems. The JSM is a JavaCard processor supporting all JavaCard bytecodes. The JSM is a fully synthesizable 32bit processor soft core with a very small footprint. The capability of it's integration in small embedded and automation systems is outlined. Special target platform is the SmartDev system which consists of a Java core interfacing to a wide variety of peripherals. SmartDev is intended to be used in mobile embedded systems for administrational, controlling and measurement purposes.**

I. INTRODUCTION

Smart cards, and hence Java Cards, are a special category of mobile embedded systems. Due to that fact, the Java Silicon Machine (JSM) introduced here meets the requirements of a wide range of embedded systems. The JSM brings the following advantages to embedded applications:

- Simplified application design and hence shorter development cycles by the object oriented Java architecture,
- Higher performance and lower resource requirements versus a JVM,
- Flexible and extensible design to natively integrate new functionality (JVM extensions, peripherals),
- Limited real time support.

The Java architecture has been designed to run its bytecode in a virtual machine that uses the functionality of the underlying operating system to manage hardware like memory or peripherals. In the JSM, it is the task of the processor to provide that functionality. New bytecodes were introduced to access the IO-Unit. That makes it easy to extend the IO-Unit with new devices commonly found in embedded systems like CAN controllers or UARTs. Fig. 1 shows a comparison of the software and hardware Java architecture.

The Java Card specification does not require the presence of a garbage collector. The lack of garbage collection results in special constraints in application design and makes the system real time capable. Hence, such systems may be applied in time critical applications.

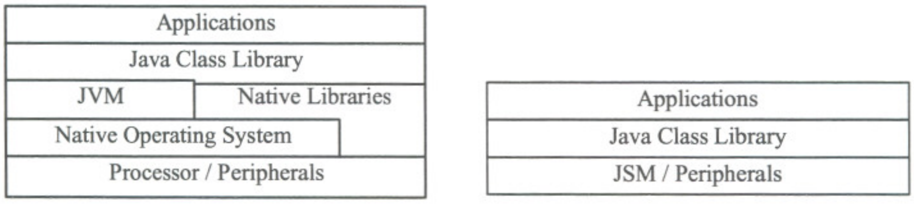
Furthermore a Java card virtual machine does neither have a class loader nor a byte code verifier as online parts, because of the limited hardware resources. These functions are executed offline. Bytecode verification and class loading are done outside the card prior to downloading the applications (cardlets) to the card's memory.

In the case of a Java Card that's not really a loss of functionality because the java card is designed with this poor communication capabilities. Usually, each data transfer is a host initiated request-response sequence.

The situation changes if the Java Card VM is used in embedded systems that may communicate via field busses or even LANs. Here, it could be desirable to load external Java code into the local Java machine to get new functionality or to interface to network services.

A wide spread embedded system architecture is that of the Intel8051 8bit micro controller. It is applied to smart cards as well as sensors or actuators. To make these legacy systems Java capable the architecture is left unchanged and a JVM is set on top of the embedded operating system. Doing so, often leads to a very poor performance of Java applications. To avoid that, the JSM could be applied to systems that have to offer both Java capability and high performance.

Furthermore, the extensibility of the JSM offers new usage scenarios. Currently it is equipped with a DES encryption unit and others are under development.



A: Software Java Architecture.

B: Hardware Java Architecture.

Fig. 1

0-7803-7241-7/01/$10.00 (c)2001 IEEE

TABLE I
DIFFERENT HARDWARE SOLUTIONS FOR JAVA AND JAVA CARD VIRTUAL MACHINES

	JSM [7, 3]	**SmartJ** [17]	**TINI^{2}J** [1]	**JSMART** [13]	**JSTAR** [13]	**LavaCore** [4]	**Jazelle** [2]	**Komodo** [11]	**MOON** [12]
Target	Softcore, Xilinx FPGA, ASIC	Integration with ST22 core	Integration with RISC processor, CPLD,ASIC	Softcore	Softcore	Softcore, Xilinx FPGA	Integration with ARM	Softcore, Xilinx FPGA	Altera APEX
Special feature	Hardware security features, Integration of external devices, e.g. I^{2}C, TDES, Support of JVM stack model	Hardware security features, Combi-nation with micro-controller	Hardware security features, Interrupt control interface	Co-processor	Co-processor	Optional extendable with SPU, DES, GC	Compatible with ARM, Combi-nation with ARM-controller.	Multithreaded Kernel., Interrupt Service Threads	Support of JVM stack model, Co-processor and Single processor solution
Support of additional codes	New bytecodes	Bilingual	Bilingual	Bilingual	Bilingual		Trilingual		
Standard	Javacard	Javacard	Javacard	Javacard	Java	Java	Java	Java	Java

II. RECENT WORK

During the last years some work has been done in developing different kinds of Java processor cores. Table I summarizes different hardware solutions for Java systems.

III. DESIGN OF JAVA CARD-PROCESSORS JSM (JAVA SILICON MACHINE)

The JSM has a modular design (see Fig. 2). The JSM processor architecture was designed to fulfill the demands of different kinds of security applications (high security design). One of the most valuable features is that the core is extendable. Different types of external devices
may be combined with the core using a simple interface.
The processor consists of the following main modules (see Fig.2).

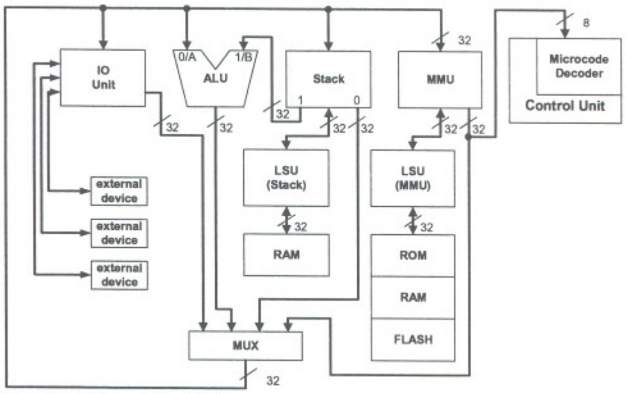


Fig. 2 Design of the JSM-processor

- Control-Unit (CU),
- Arithmetic Logical Unit (ALU),
- Memory Management Unit (MMU),
- Stack, and
- Input/Output-Unit (IO-Unit) (see Fig. 3).

A.IO-Unit

The JSM was intended as a JavaCard processor with the capability to connect external devices and for the use in embedded systems. The IO-Unit serves as an interface between the processor and the corresponding IO-devices which are not defined in the Java standard or can be accessed only by using an appropriate application programmers interface (API). Especially in a JavaCard environment the IO-Unit is necessary for the communication between the card itself and the card reader. Furthermore, the IO-Unit may be used to control additional hardware to accelerate internal computations (e.g. encryption engine). The interface for the attachment of external devices is shown in Fig. 3. Further details about structure, signals, and components of the IO-Unit as well as an example for the integration of an external device (I^{2}C-device) are explained in [3, 7].

IV. SMARTDEV(ICES)

SmartDev [8] is a Java enabled embedded platform with numerous communication facilities and interfaces to wireless and biometric technologies, positioning and security systems. It's mainly intended to be used in mobile systems and in stationary systems that require wireless access and a high level of security. SmartDev application scenarios range from mobile points of sales (POS) via access control systems to almost any remotely accessible control an measurement system.

Currently, SmartDev works with GSM wireless communication having the possibilities to exchange data over TCP/IP, SMS, FAX or simple serial connection. It's equipped with a smart card reader offering the use as POS or access controller. An enhancement of the security features is the attachment of biometric sensors. The interface for a fingerprint sensor is currently under development. Furthermore, it has a GPS unit attached that is mainly intended to be used to get positioning information in mobile devices. A side effect of the GPS usage is the presence of accurate time information that may be used in securing data transfers.

The whole system is currently built around the TINI (Tiny INternet Interface) [6], a low cost Java capable embedded system from Dallas Semiconductor. TINI's core is a DS80C390 [5], which represents a high performance Intel8051-derivate. The TINI module itself provides 10BaseT Ethernet, I2C, CAN, RS232, 1-Wire and a 4bit wide digital control port. Keyboard and display may be attached via I2C. These features make it suitable to be integrated in a broad range of systems to act as administration, control, or measurement interface (Fig. 4)

The architecture of the TINI module corresponds to that shown in Fig. 1A. On top of it's native operating system (TINI OS) work one or more JVMs. A JVM runs the bytecode directly out of the file system that lies in the battery backed RAM. Class loading and verification is done offline prior to application downloading. With the exception that the TINI's JVM and class library is partly compatible to the Java 1.1 standard this architecture is similar to that of the Java Card specification.

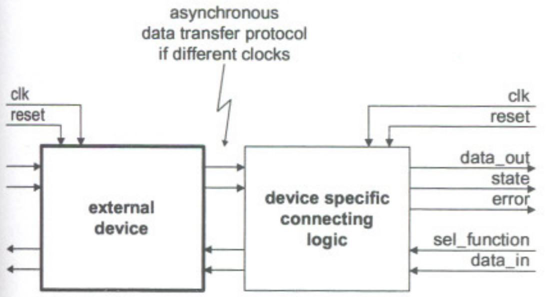


Fig. 3: Interfacing external devices

V. SMARTDEV SUPPORTED BY JSM

Because of similar functionality the TINI may be easily replaced inside the SmartDev system by the JSM making the core more lean with higher performance. The JSM's modular design allows it to integrate almost any peripheral unit on the chip and in this way to customize the chip to the application's needs. That has the effect that the class library may directly operate on top of the hardware (no drivers or native libraries are needed). Performance and reliability are improved because of the reduction of the numbers of software layers. Furthermore, the overall size of the system is reduced because external components may now be integrated.

Currently, the replacement of the TINI module by the JSM is under development. In the first step of this process the JSM core is extended by external components to provide the functionality that's not available on chip. Most important are serial communication lines to interface to GSM, GPS, card reader, and fingerprint sensor. For an easy integration into the target environment LAN and CAN interfaces are necessary. In the second step all these peripherals will be put on the JSM chip making the system compact and small and hence applicable to almost any target device.

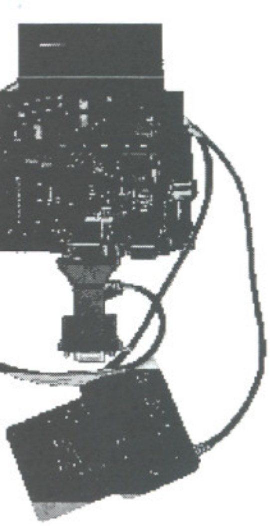


Fig. 4 SmartDev built around the TINI module

VI. FUTURE WORK

With the ongoing developments in the field of ubiquitous computing the demand for high performant embedded Java systems will increase significantly. They'll drive devices like PDAs, cell-phones, smart badges, as well as automation appliances. The functionality of the JavaCard VM is not suited to meet all these devices' needs. That's why, the JSM development will go on to provide Java1.1 compatibility and features like garbage collection, dynamic class loading, and bytecode verification. On top of these developments the new JSM-powered SmartDev architecture will be enabled to work in Jini [14] environments.

VII. REFERENCES

[1] Advancel Logic Corporation, Tiny2J Micro*processor Core for JavaCard Applications,* http://www.advancel.com

[2] JazelleTM – ARM® Architectumre Extensions for Java Applications, White Paper, www.arm.com

[3] N. Bannow, Java-processor for SmartCards and small embedded systems. (in german) Diploma thesis, Institute of Applied microelectronics and computer engineering, University of Rostock, Dec. 2000

[4] Bhaskar Bose, M. Esen Tun; LavaCORE - A Configurable Java Processor; http://www.xilinx.com/xcell/xl37/xcell37_20.pdf

[5] Dallas Semiconductor. DS80C390 Dual CAN High-Speed Microprocessor, 1999. http://www.dalsemi.com/datasheets/pdfs/80c390.pdf

[6] Dallas Semiconductor. DSTINI1 TINI Verification Module, 2000. http://www.dalsemi.com/TINI/dstini1.pdf

[7] H. Ploog, R. Kraudelt, N. Bannow, T. Rachui, F. Golatowski, D. Timmermann: A Two Step Approach in the Development of a Java Silicon Machine (JSM), Workshop on Hardware Support for Objects And Microarchitectures for Java. Austin, Texas, October 1999

[8] T. Geithner, Mobile GSM Powered Web Server. (in german) Project Work, Chair of Information and Communication Services, University of Rostock, May 2001

[9] F. Golatowski, H. Ploog, R. Kraudelt, T. Rachui, O. Hagendorf: Java Virtual Machines for resource critical embedded systems and SmartCards,. (in german), Java Informationstage JIT 99, ITG/GI-Fachtagung, Düsseldorf, September 1999

[10] F. Golatowski, N. Bannow, H. Ploog, J. Hildebrandt, D. Timmermann: JSM- Java Processor for embedded systems: Design, Implementation and Rapid-Protoptyping, (in german), 10. Maritimes Symposium Rostock, June 2001

[11] J. Kreuzinger, R. Zulauf, A. Schulz, Th. Ungerer, M. Pfeffer, U. Brinkschulte, C. Krakowski, Performance Evaluations and Chip-Space Requirements of a Multithreaded Java Microcontroller, Workshop on Hardware Support for Objects And Microarchitectures for Java. Austin, Texas, October 2000

[12] Moon Java Processor Core, http://www.vulcanasic.com/eda.htm

[13] Java Coprocessor, The JSTAR, Product brief, http://www.nazomi.com

[14] Sun Microsystems. Jini Network Technology, 1998. http://www.sun.com/jini/

[15] Sun Microsystems, Inc., Java Card Technology Home Page, http://java.sun.com/products/javacard

[16] Sun Microsystems Inc., Java Card TM 2.1.1 Virtual Machine Specification, 2000

[17] SGS Thomson, Instant Java for your smartcard, http://www.st.com/stonline/prodpres/smarcard/insc9901.htm