Middle East Technical University
Department of Computer Engineering

# Kobot: A mobile robot designed specifically for swarm robotics research

Ali E. Turgut, Fatih Gökçe, Hande Çelikkanat, Levent Bayındır and Erol Şahin

METU-CENG-TR-2007-05

November 2007

Department of Computer Engineering
Middle East Technical University
İnönü Bulvarı, 06531, Ankara
TURKEY



Technical Report

This page contains a Turkish translation of the title and the abstract of the report. The report continues on the next page.

---

# Kobot: Sürü robot çalışmaları için tasarlanmış gezgin robot platformu

Ali E. Turgut, Fatih Gökçe, Hande Çelikkanat, Levent Bayındır and Erol Şahin

Bilgisayar Mühendisliği Bölümü
Ortadoğu Teknik Üniversitesi
İnönü Bulvarı, 06531, Ankara
TÜRKİYE

## Öz

Bir sürü robot sisteminin parçası olacak bir robottan beklenecek özellikler, tek başına çalışması hedeflenen bir robotunkilerden farklı olacaktır. Bu çalışmada, sürü robot araştırmalarında kullanılacak ideal bir robottan beklentilerimizin neler olacağı tartışılmış ve bu beklentileri mümkün olduğunca karşılayabilmek amacıyla geliştirdiğimiz Kobot platformu incelenmiştir. Platformun genel tasarımının ardından, ortamın ışık koşullarından ve çevredeki diğer robotlardan az etkilenmesi için özel olarak tasarlanan bir kızılötesi kısa mesafe algılama sistemi anlatılmıştır. Algılama sisteminin başarımı, yapılan sistematik deneylerle değerlendirilmiştir. Ayrıca, Kobot platformunun IEEE802.15.4/ZigBee tabanlı haberleşme sistemi ve bu sistemin üzerine kurulmuş, robotların teker teker ya da paralel olarak programlanmasına olanak sağlayan kablosuz programlama sistemi anlatılmıştır. Platformun başarımını ölçebilmek için, Kobotlarla sürü şeklinde hareket etme davranışı gerçekleştirilmiş ve bu davranışa ait görüntüler sunulmuştur. Son olarak, şu ana kadar sürü robot sistemleri için geliştirilmiş, ya da bu araştırmalarda kullanılmakta olan robot platformları incelenmiş ve aranan özellikler bakımından Kobot platformuyla karşılaştırılmıştır.

# Contents


**1 Introduction** **1**

**2 Wishlist for Swarm Robotic Systems** **1**

**3 The Kobot Robotic System** **3**

**4 The IR-based Short-Range Sensing System** **5**

4.1 Sensing . . . . . . . . . . 5

4.2 Coordination . . . . . . . . . . 5

4.3 Characteristics . . . . . . . . . . 6

**5 The Communication System** **7**

**6 Kobot Simulator and Flocking with Kobots** **10**

**7 Review of mobile robots developed for, or being used in, swarm robotics research** **10**

**8 Conclusions** **11**

## Abstract

The requirements of a mobile robot to be used as part of a swarm robotic system differs from that of a mobile robot to be used as stand-alone. In this paper, we first provide a wishlist of requirements that would be sought for in mobile robot platforms to be used in swarm robotics research. Then, we describe Kobot, as a new mobile robot platform which is designed to satisfy as much of these requirements. Specifically, we first describe in detail, an infrared-based short-range sensing system that can make proximity measurements with minimal interference from environmental lighting conditions as well as from other robots. The performance of the system, is evaluated with systematic experiments. Then, we present an IEEE802.15.4/ZigBee-based communication system, which is used to develop a system, that can wirelessly program robots either one-by-one or in parallel. Finally, we provide snapshots from the flocking of a Kobot robotic swarm. The paper, reviews and evaluates existing robot platforms that are developed for, or being used in, swarm robotics research in comparison with the Kobot platform and concludes.

## 1 Introduction

Swarm robotics [1] is a new approach to the coordination of large numbers of robots that takes its inspiration from the impressive coordination abilities of social insects. It studies how a large number of robots can interact to create collectively intelligent systems without any centralized coordination and achieve robustness, flexibility and scalability at the system level.

There exist two challenges for swarm robotic systems towards their use in real world applications. First is the need for large numbers of robots, which settles for no less than the means of well-established mass production. Second is the need for robust, flexible and scalable coordination methods to operate on swarm robotic systems. The latter challenge, requires that a swarm robotic platform be available for research, that would facilitate study, rather than interfering with it, and allow the researchers to concentrate on the problems of coordination. The requirements of a mobile robot to be used as part of a swarm robotic system differs from that of a mobile robot to be used as stand-alone. It takes more than gathering a number of copies of any robot platform and make them work together since there exist a number of additional constraints, which follow the nature of swarm systems.

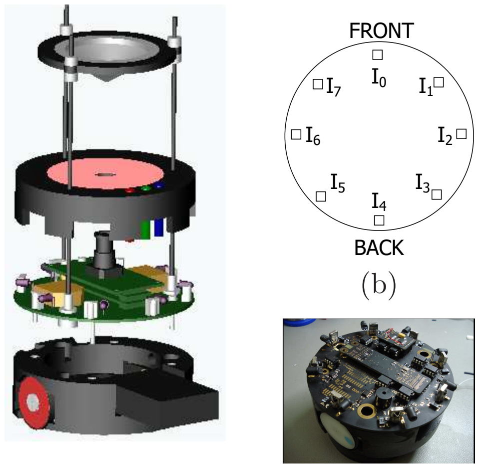
In the next section, we will first discuss the additional requirements that are needed in robots that would be used in swarm robotic systems, as a research wishlist. Then we will present a new mobile robot that is designed from scratch to meet most of these requirements. We will discuss a novel short-range infrared sensing mechanism that can address some of the constraints in section 4. Following that, communication system which is used for the wireless parallel programming of robots will be explained in section 5. The next section, section 6, will present a case study, in which we exploit the presented short-range sensing capabilities for a flocking behavior, both in the simulation environment, and on real robots. We will finally review existing robot platforms, comparing them with the Kobot robotic system, in section 7.

## 2 Wishlist for Swarm Robotic Systems

The requirements of a mobile robot to be used as part of a swarm robotic system differs from that of a mobile robot to be used as stand-alone. In this section, we investigate in detail the extra requirements expected from robots that would be used in swarm robotic systems.

- **Sensing and Signalling**: The main emphasis in swarm robotics is the interaction among the robots as well as the interaction of the robots with their environment, putting extra constraints to the robots to be used.
  - **Interference among robots**: The interference among the sensing systems of the robots should be minimal. In most stand-alone robot systems, the proximity sensors are not designed to handle interference that may result from other robots operating in the same environment.

- **Interference from environmental factors**: The interference of environmental factors with the sensing system of the robots should be minimal. Existing stand-alone robot systems the effect of environmental factors, such as the effect of daylight on the infrared proximity sensors, are accepted as part of the world, and not dealt with. In swarm robotics research however, a major emphasis lies on the use of self-organizing coordination methods, and that environmental non-uniformities which may bias the experiments would be unacceptable.
  - **Kin-detection**: The robots should be able to distinguish other kin-robots. In most stand-alone robot systems, such an ability is usually regarded as high-level and is usually handled through visual processing. In swarm robotics systems, however, such a sensing ability is regarded as essential to study coordination mechanisms involved in tasks such as flocking or pattern formation. Therefore, it is preferred to do kin-detection as easy as proximity sensing.
  - **Stigmergic sensing and signalling**: The robots should be able to leave "marks" in the environment and be able to sense them. This is a rather difficult and challenging task for even stand-alone robotic systems. Although these abilities can be considered as exotic for a stand-alone robotic system, they are high on the wishlist, since stigmergic communication is known to be heavily used by social insects for coordinating their behaviors.
  - **Generic sensing**: The robots should also provide some form of generic sensing to allow the researcher to test novel sensing strategies, which need not be present in the existing sensing abilities that are implemented by the fixed hardware design.
- **Communication**: Unlike stand-alone robotic systems, communication by plugging cables to the robots are no longer feasible when working with a swarm robotic system and therefore the robots have to support some form of wireless communication.
  - **Wireless communication between the robots and a console**: The robots should support wireless communication with a console, to allow easier monitoring and debugging of algorithms on individual robots.
  - **Wireless communication among robots**: The robots should have inter-robot wireless communication. Such an ability would allow the robots be used in the formation of ad-hoc networks.
  - **Wireless programming**: The robots should support wireless downloading of control algorithms from a console.
  - **Parallel programming**: The robots should be programmable in parallel. In swarm robotics research, the robots usually share the same control algorithms and programming the swarm as a whole in one-shot would be a big time-saver.
- **Physical interaction:** The robots should be able to physically interact with each other and the environment. Self-assembly or self-organized building of objects into large structures remain an interesting research topic in swarm robotics.
- **Power:** The robots should have a long battery life. In swarm robotics research, the swarm may need to run around for a period that is long enough for the collective behaviour to emerge, and the goal to be reached. Low battery life would imply that many robots would run out batteries that need to be recharged and replaced frequently.
- **Size**: Size does matter in swarm robotic systems. The robots should be small enough not to increase the size of test arena when experimenting with the system, and yet big enough not to limit the expandability of the robot or increase the cost of the swarm robots due to miniaturization in components.
- **Cost**: The robots should be cheap as much as possible, since unlike stand-alone robots, they will be sold at least in groups of tens.
- **Simulation:** The swarm robotic systems require realistic simulators which would be essential to speed up development of new control algorithms. Such simulators need to model the




(a) (b) (c)

Figure 1: (a) The exploded view of Kobot, equipped with the optional omni-directional vision system. (b) The positioning of the IR sensors, and their numbering. (c) The basic version of Kobot. The cap of the robot is removed, exposing the short-range sensing board and the main controller and the wireless communication boards stacked on top.

interactions between the robots as well as the interactions of the robots with their environment in a realistic way that is also verified against the physical robots.

Satisfying all of these constraints in a single design is a difficult, if not impossible, challenge. The design choices made regarding one requirement, such as size, poses additional constraints towards the reaching of other requirements such as power and communication. Hence, the design process should take all of these constraints at once and try to find an optimum design solution.

In this paper, we introduce a new robot platform that is designed to tackle most of these requirements. In the next section, we will discuss the design choices related to the requirements listed above.

## 3 The Kobot Robotic System

Kobot is a circular differential-drive robot whose design is mainly guided by the constraints discussed in the previous section, Figure 1. Kobot, with a diameter of 120mm (the size of a CD) and a weight of 350 grams with batteries, is designed to be a light, small, yet extendable and power-efficient and relatively cheap robot platform for swarm robotics research.

Kobot's body consists of two pieces: (1) a cylindrical base, which houses the motors, the battery pack and the short-range sensor board, and (2) a cylindrical cap that covers the body. Both pieces are manufactured by casting polyurethanes, a low-density material that is ideal to create a light yet durable body structure for the robot. The cap is wrapped with white paper, to increase the "visibility" of the robots to each other. We used high efficiency (to save power), low profile (to save space), high torque DC gear-head motors from FTB Inc. of Faulhaber which are directly connected to the wheels. The motors are driven using Vishay Inc.'s high switching frequency Si9988 motor drivers.

The overall system design of Kobot is shown in Figure 2. At the heart of the Kobot, there is the control sub-system to which all of the information from the other sub-systems, that is short-range sensing, communication, vision and power, are fed to the control sub-system. With the information acquired, a 20MHz PIC16F877A, called the main controller, implements the control algorithm which determines the behaviors of Kobot.

In its basic version, Kobot comes with a novel IR-based short-range sensing sub-system shown in Figure 1(b) and 1(c). The system is custom-designed to address the first three concerns of sensing as discussed above. It addresses kin detection problem through the sensor-level detection

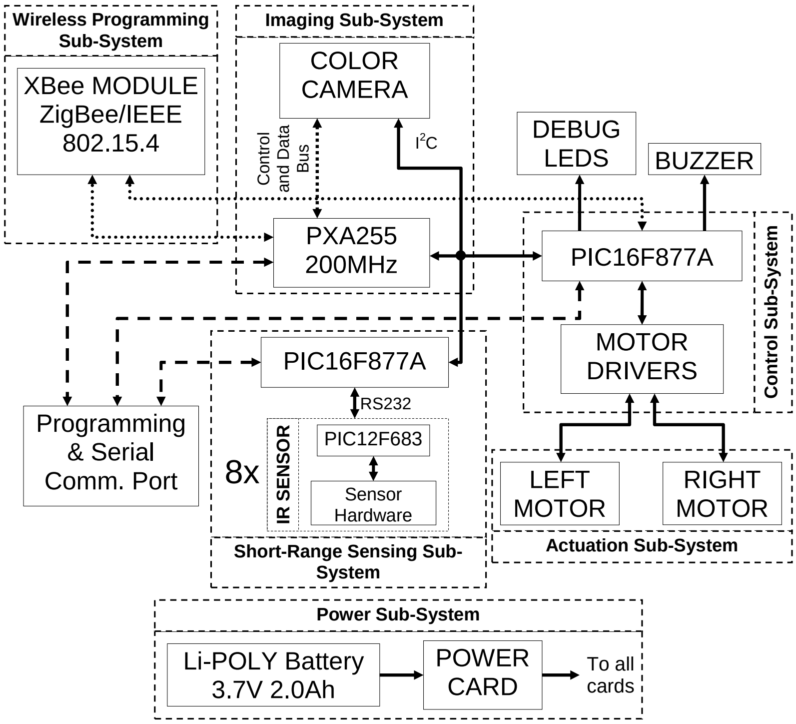



Figure 2: Schematic diagram of Kobot's control system.

of kin-robots from other objects, minimum robot interference by utilizing a sensor-level strategy which performs the measurements only when there is no sensor detected around the radiating IR signal, and minimum environmental interference by employing modulated IR signals. This sub-system also hosts a buzzer and three LED's whose lights are transmitted to the top of the cap using circular light guides to help debugging.

Kobot provides wireless support using the IEEE802.15.4/ZigBee protocol. This protocol provides a low-power networking capability that can support point-to-point, point-to-multipoint and peer-to-peer communication. This protocol, was preferred to IEEE802.11 over its power efficiency, and to Bluetooth over its ability to address 65536 modules instead of 7 as supported by Bluetooth. Through this system wireless communication between robots, between robots and a console is supported. Also, using the broadcasting ability of this protocol, we were able to implement wireless parallel programming of multiple robots.

The basic version of Kobot is planned to be extendable by a general-purpose omnidirectional vision sub-system as seen in Figure 1(a). This system is composed of a camera facing an omnidirectional mirror placed on the top of the Kobot. It can view a region of $0.9m$ radius, shrinking the view with a constant proportion, independent of the distance. The vision subsystem contains a 200MHz PXA255 microprocessor, which runs the Linux operating system on 32MB EEPROM program memory and 32MB RAM providing a powerful research platform for the on-board execution of machine vision algorithms in nearly real-time. Although designed and implemented as a power-efficient standalone system, the vision system has not yet been fully integrated into the Kobot and will not be discussed in the rest of the paper.

On the overall, the Kobot is designed to be a modular system. The short-range sensing system acts as the "main board" of the robot onto which other electronic cards are mounted using small-form connectors and communication among different cards and controller uses the $I^2C$ protocol.

The issue of the power efficiency, has been one of the main concerns of our design and has influenced the design at all aspects, from the choice of the body material to the design of the electronics and the choice of motors. As a consequence of this, our preliminary experiments showed that Kobot (performing obstacle avoidance in a closed environment, with the ZigBee module set in receive mode) has an operation duration of 7.5 hours when using 3 AA-sized NiMH batteries (2700mAh each), and 10 hours with a one-cell Li-Poly battery (2000 mAh).

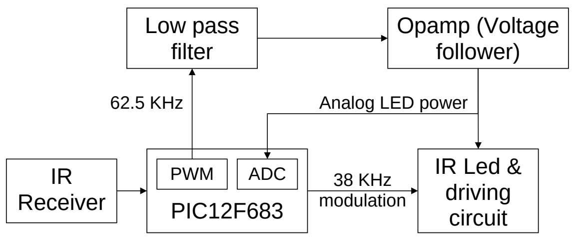



Figure 3: Block diagram of an individual sensor.

# 4 The IR-based Short-Range Sensing System

We have designed a novel low-power IR-based sensing system shown in Figure 1(c). The system consists of 8 sensors mounted on a circular board 45° apart from each other, whose operation is coordinated by the main sensor controller, 20MHz PIC16F877A.

## 4.1 Sensing

The basic structure of each sensor is illustrated in Figure 3. An IR led (TSAL7400 from Vishay) with a half cone angle of 25° is used as the transmitter. A standard 38KHz modulated infra-red receiver is used in order to eliminate the interference from the light sources in the environment. A low pass filter and an op amp configured as voltage follower are employed to produce an adjustable power for the infra-red led using PWM module of the controller. The led power is also monitored by using ADC module of the controller. The sensor's operation is carried out by a Microchip PIC12F683 which includes PWM and ADC modules in a small package and has an internal 8MHz oscillator.

A sensor's execution iterates over three states; *kin-detection*, *proximity-sensing* and *data-transmission*. The transitions among these states are controlled by the main sensor controller. The main sensor controller is also responsible for updating the debugging led of each sensor, which indicates the presence of an object in the sensing range.

In the *kin-detection* state, the sensor scans the environment to detect if there are any modulated IR signals. A detected signal indicates the existence of a kin-robot in that direction. In order to remove the possibility of cross-talk with the other sensors on the robot, the main sensor controller keeps all the sensors "silent" during this phase. The *proximity-sensing* state begins with an initial scan of the environment in order to seize an opportunity to begin its measurements, without interfering the others' measurement. In order to ensure that there are no other sensors making simultaneous measurements, the sensor has to wait for a certain time(6.7ms) without receiving any IR signal, which is the maximum possible time between two consecutive IR signals. Then, the sensor adjusts the led power to one of the 7 levels, sending an IR signal at each level, to determine the minimum level at which the radiated signal reflects back from an obstacle. This minimum level determines the distance of the obstacle. Since there are 7 power levels, the sensor determines the distance information as in one of the 7 discrete levels. In the *data-transmission* state, the sensor sends the measurements to the main sensor controller via software serial communication.

In Figure 4, the working of a sensor over these three states is illustrated as a timing diagram for detecting an obstacle at *Level* 3.

## 4.2 Coordination

In order to minimize the possibility of cross-talk among the sensors of a robot, a centralized coordination mechanism is needed. This coordination is performed by the main sensor controller which controls the states of the sensor using the interrupt features of sensor controllers and the software implemented serial communication. The controller begins by initializing all the sensors to the *kin-detection* state. The duration of this state is assigned to a random time slot between 11ms and 15ms to prevent any possible synchronization between robots. Otherwise, synchronization can cause the robots to continue listening to each other without radiating any signals, which prevents them from sensing each other. On the expiration of the random time slot, the controller switches

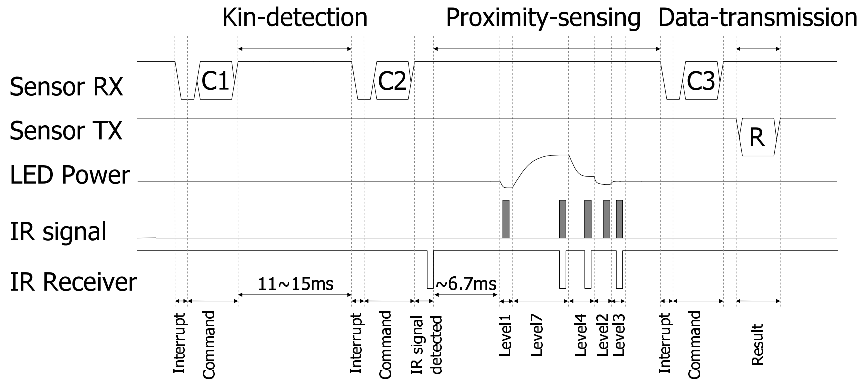



Figure 4: Timing diagram of an individual sensor is illustrated for detecting an obstacle at *Level* 3. The sensor is first initialized to *kin-detection* state with the command sent by the main sensor controller after the interrupt and it starts scanning the environment to detect any kin-robots. No IR signal is received during this state, meaning that there is no robot in the sensing range. The sensor will then try to detect the obstacle and measure the distance. In the *proximity-sensing* state, the sensor begins with scanning the environment to detect any sensors making simultaneous measurement. The sensor receives a signal and waits until the other sensor finishes the measurement. Since no IR signal is received during 6.7ms (maximum possible time between two consecutive IR signals), the sensor starts its measurement by varying the led power to determine the minimum level at which the radiated signal reflects back. After the third interrupt, the sensor is switched to *data-transmission* state and sends the result to the main controller.

the even numbered sensors, $I_0$, $I_2$, $I_4$ and $I_6$ shown in Figure 1(b), into the *proximity-sensing* state. The other sensors are delayed for 2ms. Without this delay, two neighboring sensors might start the measurement almost at the same time after waiting 6.7ms without receiving any signals, if there is no robot around. In such a case, the signal from one sensor might reflect from an obstacle and be detected by the neighboring sensor, resulting in a false reading. The controller then collects the measurements from the sensors and updates the debugging leds accordingly. One such control step takes about 55ms, resulting in an update rate of approximately 18Hz. The results are sent to the main controller via $I^2C$ protocol at 400kHz upon request.

### 4.3 Characteristics

Two different experiments are performed for evaluating the performance of the short-range sensing system. In the experiments, one robot is responsible for making the measurements, and placed at the origin of the coordinate system with its sensors turned on. A second robot is moved in the first quadrant of x-y plane, with 1 cm increments while its sensors are turned on or turned off according to the experiment (Figure 5(a)).

The first test performed is the range test of obstacle detection. The sensors on the moving robot are closed and it acts as an obstacle. Each time it is moved to a new location, 200 samples are collected by the front sensor. The results of the experiment are illustrated in Figure 5(b), where the gray value of a region indicates the average measured proximity of the obstacle for the corresponding location. The sensor has a range of approximately 21cm in lateral direction and 10cm in transverse direction, with a variance of approximately 1cm. The performance of the sensor in this test is found to be acceptable.

The second scenario aims to evaluate the performance of robot detection. Using the same setup and turning on the sensors of the second robot, it is moved without changing its orientation with respect to x-y plane. The success rate of robot detection is given in Figure 5(c) where the gray value of a region indicates the average percentage of success rate for the corresponding location over 200 samples. The performance in robot detection is quite satisfactory at long distances (between 3-25cm), and decreases dramatically in short distances(within up to 3cm) which is not

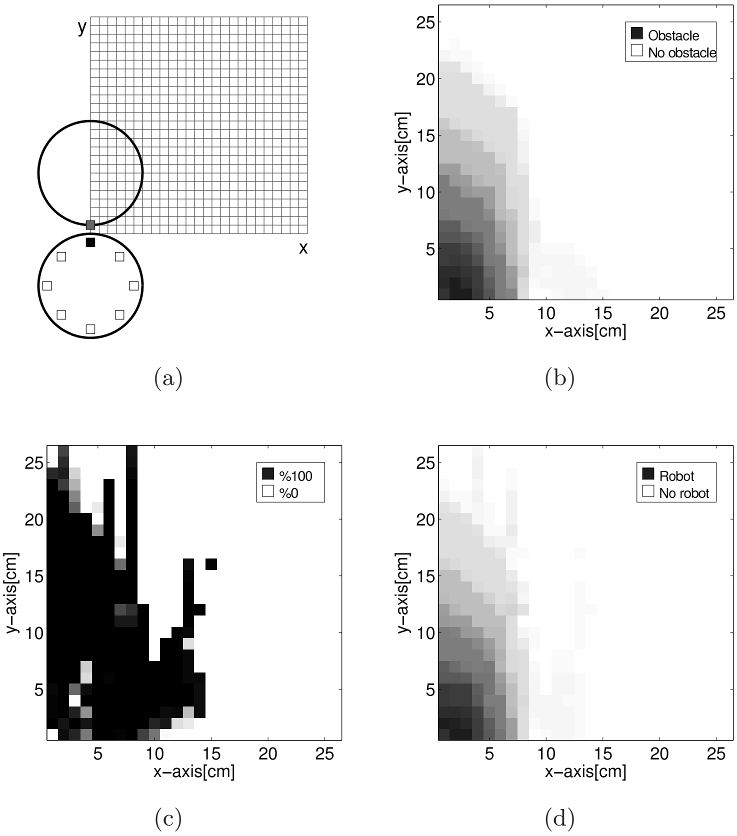



Figure 5: (a) The experimental setup for the experiments is sketched. The circle at the left lower corner indicates the robot which is used to make the sensing measurements. Samples are collected from the front sensor while the other sensors are also turned on. For each of the grid positions shown in the figure, a circular object (be it an obstacle or another robot) is placed, and the sensor readings are recorded. (b) The gray scale image drawn according to the average range values obtained from the range test of obstacle detection. (d) The gray scale image drawn according to the average range values obtained from an active robot. (c) The gray scale image shows the success rate in the detection of a robot.

actually a problem since robots are not planned to operate in this close proximity. Performance sometimes decreases at certain points in the operating range which might be due to interference from the several sensors of the measured robot. The average distance values for this experiment are illustrated in Figure 5(d) as gray level patches. By comparing these average distance values with the ones in Figure 5(b), we can conclude that robot-to-robot interference of the system is also acceptable.

# 5 The Communication System

The communication system is designed to facilitate communication of robots with a console, parallel programming and communication among robots wirelessly.

In the Kobot system, Maxstream's XBee module is used as the communication hardware. XBee modules are IEEE802.15.4/ZigBee protocol compliant which support point-to-point, point-to-multipoint and peer-to-peer communication, as well as mesh networking[4]. XBee modules communicate with the host via serial interface up to 115.2kbps and provides a 250kbps RF data transmission rate for wireless communication. We have utilized XBee modules at a data rate of 57.6kbps.

Algorithm 1: Parallel Programming algorithm

```
{User marks the Kobots to be programmed}
marked Kobots are put in CandidateList
{User selects the program to be downloaded}
load the program to be downloaded
for Robot in CandidateList do
  send ENQ to Robot
  if not timeout and reply equal to ACK then
    change robot's tag to present
  else
    change robot's tag to not present
  end if
end for
for block in program do
  for Robot in CandidateList do
    if there exists RetryTaggedRobot then
      put all RetryTaggedRobot in ProgramList
      change block to previous
    else
      put all PresentTaggedRobot in ProgramList
    end if
  end for
  for Robot in ProgramList do
    send ENQ to robot to request checksum result
    wait for ACK from robot
    if no reply or unexpected character then
      change tag to notpresent
    else
      if reply FALSE and retry less than 3 then
        change tag to retry
      else
        change tag to present
      end if
    end if
  end for
  broadcast START character
  broadcast block length
  broadcast starting memory address
  broadcast checksum
  broadcast program block
end for
broadcast END character
```

On this communication hardware, we have implemented the parallel programming feature. The software of parallel programming consists of two parts: One is the host program running on a console which interacts with the user to conduct the downloading procedure. The other is the bootloader program running on the main controller of each Kobot. It initiates the downloading process after boot-up if requested by the host program. During this process, the program memory must be re-written, which is a feature supported by the main controller PIC16F877A.

PIC16F877A has various open source bootloaders available. One of the most popular of them is the Screamer of Sparkfun Inc. Since Screamer does not support wireless or parallel programming, we have extended its capabilities accordingly. The bootloader is ported to PIC-C C-compiler of CCS Inc., and modified to enable the use of watchdog timer and interrupts. Remote control facilities for starting and stopping the Kobots are also added to the host program of the Screamer.

The algorithm of the host program is shown in algorithm 1. In a typical scenario, the user first stops currently running Kobots, then marks the Kobots to be programmed and selects the program to be downloaded. The selected program is splitted into blocks of 4/8 words long. The blocks are fetched in sequence, their block lengths, starting memory addresses and checksums are determined. These values are broadcasted and checksum verification is performed on each robot individually.

Meanwhile, Kobots are kept in a wait state, Algorithm 2. On receiving the last byte of the program block, they calculate the checksum. If it is correct, they write the block to their program memory and reply as TRUE. Otherwise, the write operation is not performed and FALSE reply is sent. Then the host program re-broadcasts the same block. This procedure continues until all the blocks are broadcasted and received successfully.

Algorithm 2: Bootloader algorithm

```
if not timeout and ENQ received then
  send ACK to console
else
  run current program
end if
while TRUE do
  wait for ENQ
  if checksum false then
    reply FALSE
  else
    reply TRUE
  end if
  wait for START character
  receive block length
  if block length equal to END then
    start downloaded program
  end if
  receive memory address
  receive checksum
  receive program block
  calculate checksum
  if checksum = 0 then
    write program block to memory
  end if
end while
```

## 6 Kobot Simulator and Flocking with Kobots

Kobot features a physics-based simulator based on the ODE (Open Dynamics Engine), which enables the modeling of actual physical interactions by implementing physical concepts such as friction and collision. It is calibrated against the physical robots using the results of systematic experiments. The results of the IR sensor characteristics, as presented in section 4, is used to implement the sensors in the simulator.

As a case study, we have implemented a crude flocking algorithm with seven Kobots. Flocking is a problem that can be implemented easily in systems where each agent can perceive the relative position and orientation of its neighbors. However, it becomes a challenging problem when robots do not identify their neighbors, and act upon anonymous proximity sensing. As a result, flocking depending on proximity sensing only is a good benchmark test for evaluating the performance of the new IR-based sensing system. Figure 6 presents the flocking behavior on the Kobot simulator, and a scene of the real robots running the same flocking behavior[1].

## 7 Review of mobile robots developed for, or being used in, swarm robotics research

In this section, we review the existing mobile robot platforms that are developed (or can be used) for conducting swarm robotics research and evaluate them according to the wish-list listed in Section 2

**e-puck**[3] has a circular shape with a diameter of 70mm and is made of plastic. Two stepper motors are used for locomotion and there is a speaker for audible feedback. An accelerometer, 8 IR sensors (for obstacle and ambient light detection), a camera, 3 omni-directional microphones and a Bluetooth module are utilized in robots where vision, color LED communication and ZigBee communication modules can be added on-demand. Robots can be programmed via the Bluetooth module. A three hours of autonomy is reported using a 5Wh LIon battery.

**Alice**[5] has a rectangular prism body, having dimensions of 21x21x21mm and made of plastic and PCB. Two high efficiency Swatch motors are used for locomotion. Alice has many optional sensory modules such as 4 IR proximity sensors for obstacle detection and short-range robot-to-robot communication module, IR receiving module, linear camera module, radio (RF) communication module and ANT extension module. In addition to that there is an optional gripper module and various locomotion modules for different terrains. Ten hours of autonomy is reported with two button batteries and twenty hours of autonomy is achieved with an additional LiPo battery.

**Jasmine**[2] is another micro-robot platform which has a rectangular prism body having 26x26x26 mm and made of aluminum and PCB. Two small gear-head motors are used for locomotion. Jas-

[1]For a complete demonstration of the flocking behavior on simulator and with real robots, see the video attached to this paper


Figure 6: (a)A snapshot from the flocking behavior of Kobots on the simulator (b)A scene showing the real robots performing the flocking behavior

mine has 6 IR sensors (LED+receiver) for proximity and local communication with robots. There is one powerful IR LED for detailed analysis of an object of interest and an IR communication module with host. Jasmine III has a modular design in which different sensing modules like ambient light sensor, color sensor and different locomotion modules can be utilized. Two hours of autonomy is reported with LiPo batteries.

**s-bot**[6] has a circular shape having a diameter of 116mm and a height of 100mm. s-bot's have a patented locomotion sub-system consisting of both wheels and tracks which are driven by two DC gear-head motors. s-bot's have many sensors of different modalities. These sensors, not to mention all, are 15 IR proximity sensors for obstacle detection, torque sensor on wheels, force sensor between base and wheels, 3-axis accelerometer, omni-directional camera/8 RGB LED's for messaging between each s-bot. There are two grippers for holding/lifting other s-bots and objects. There is a Wi-Fi module for wireless communication. Lifetime of batteries (Li-Ion) are not mentioned in the study.

**Swarmbot**[7] is a square shaped robot having dimensions of 130mm. It has got four wheels on each side driven by two DC gear-head motors. There are bump sensors, light sensors, drive wheel encoders and patented ISIS IR robot-to-robot communication system in a Swarmbot. Additional modules are linear CCD, magnetic food and swarm-cam emitters which can be utilized on-demand. There is an RF communication unit for debugging and programming purposes. Battery life is not reported but there are automatic charging stations in which robots can recharge themselves.

**Pherobot**[8] has circular shape with a diameter of 110mm. There are two DC motors in the locomotion sub-system. There are 8 modulated IR transmitters and receivers in each robot for robot-to-robot communication. An interesting user interface is deviced in robots in which the user wears virtual reality goggles to interact with robots. No information is available for battery life time and there is no wireless communication module in robots.

**Khepera II**[9] has a circular shape with a diameter of 70mm. It has two DC brushed servo motors with incremental encoders and 8 infra-red proximity and ambient light sensors. There are additional modules such as different camera modules(black&white or color and direct or omnidirectional), wireless video module, linear vision module, matrix vision module, radio communication module, gripper and general I/O module which can be integrated to the robot on-demand. One hour of autonomy is reported with NiMH batteries.

**Flockbots**[10] are circular shaped robots with a diameter of 180mm. Two modified RC servomotors are used for locomotion. There are 5 IR range finders, one bump sensor and a CMUCAM-2 vision module. A gripper is also designed to grip objects. Additional wireless communication modules are planned to be added to robots. A Bluetooth module is used for debugging and downloading programs. Two hours of autonomy is reported with NiMH batteries.

When compared against the existing robot platforms reviewed above, Kobot has three major strengths. First, Kobot provides a simple IR-based short-range sensing system which is designed to minimize interference from the environment as well as from other robots, while providing kin-detection ability. Second, Kobot provides support for wireless parallel programming of swarm robots through the IEEE802.15.4/ZigBee protocol. To the best of our knowledge, Kobot is the first robot system to provide this facility. Third, Kobot is the only medium-sized robot that can provide up to 10 hours of battery life which is second only to the miniature Alice robot. For a detailed comparison refer to table 1.

# 8 Conclusions

In this paper, we have first discussed the extra requirements for a robot platform to be useable in a swarm robotic system. Then, we presented Kobot, a new mobile robot which is designed to address most of these requirements. Specifically, we have presented a novel IR-based short-range sensing system which can sense kin-robots, and has minimal interference from environmental lighting and from other active robots. We have also presented a wireless communication system, and discussed the implementation of a wireless parallel programming system. Our initial experiment have shown that Kobot is a cheap yet capable platform for swarm robotics research.

There are several items that can be regarded as future work. Imaging sub-system is yet to be fully integrated, as an optional sensing system. A new short-range sensing system is planned to be designed which will be faster, more modular and will enable short-range communication between

Table 1: Comparison of Robots in the literature

| | **e-puck** | **Alice** | **Jasmine** | **s-bot** | **Swarmbot** | **Pherobot** | **Khepera II** | **Flockbot** | **Kobot** |
|---|---|---|---|---|---|---|---|---|---|
| **Interference robots** | N/A | YES | YES | N/A | NO | NO | YES | YES | NO |
| **Interference environment** | NO | NO | NO | YES | NO | NO | NO | NO | NO |
| **Kin detection** | YES | YES | YES | YES | YES | YES | YES | NO | YES |
| **Stigmergic sensing** | NO | NO | NO | NO | NO | NO | NO | NO | NO |
| **Generic Sensing** | vision | vision | NO | vision | vision | NO | vision | vision | vision+IR |
| **Wireless com.** | Bluetooth 802.15.4 ZigBee | RF modem | IR | Wi-Fi | RF modem | NO | RF modem | Bluetooth | 802.15.4 ZigBee |
| **Wireless prog.** | YES | NO | NO | YES | YES | NO | YES | YES | YES |
| **Parallel prog.** | NO | NO | NO | NO | NO | NO | NO | NO | YES |
| **Wireless robot-robot** | YES | YES | YES | YES | YES | YES | YES | YES | YES |
| **Physical Int.** | NO | gripper | gripper | gripper | NO | NO | gripper | gripper | NO |
| **Battery Life** | medium | high | medium | N/A | N/A | N/A | short | medium | high |
| **Size (cm)** | dia.= $\phi7$ | 2.1*2.1*2.1 | 2.6*2.6*2.6 | dia.= $\phi12$ hei.=15 | 13*13*13 | dia.= $\phi11$ | dia.= $\phi7$ hei.=3 | dia.= $\phi18$ | dia.= $\phi12$ hei.=7 |
| **Cost** | low | low | low | N/A | N/A | N/A | high | medium | low |
| **Simulator** | Webots | Webots | Breeve | Swarmbot3D | N/A | N/A | Webots | MASON and Breeve | CoSS |
| **Computation** | 30F6014A | 16F877 | ATMEGA18 | XScale | 40 MHz ARM | PALM V | 25MHz Motorola 68331 | PXA255 | 16F877A |

nearby Kobot's. Communication among Kobot's will also be enabled using ZigBee protocol.

## References


[1] E. Sahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application", *Swarm Robotics Workshop:State-of-the-art Survey, Erol Sahin and William Spears, editors, 3342, Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, pp.10-20, 2005.

[2] *Open-source Microrobotic Project*. University of Stuttgart. 19 March 2007. http://www.swarmrobot.org

[3] Cianci C. M., Raemy X., Pugh J. and Martinoli A., "Communication in a Swarm of Miniature Robots: The e-puck as an Educational Tool for Swarm Robotics", *Proceedings of the Swarm Robotics Workshop*, Springer-Verlag, Berlin, Germany, 2006.

[4] http://www.maxstream.net/support/knowledgebase/files/XST-AN021a-ZigBee%20and%20802154.pdf

[5] Caprari, G. and Siegwart, R., "Mobile Micro-Robots Ready to Use: Alice", *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Canada, in press. 2005.

[6] F. Mondada, G.C. Pettinaro, A. Guignard, I.V. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L.M. Gambardella, and M. Dorigo. "SWARM-BOT: a new distributed robotic concept", *Autonomous Robots*, Special Issue on Swarm Robotics, 2003.

[7] McLurkin J., Smith J. "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots", *7th International Symposium on Distributed Autonomous Robotic Systems*. France, 2004.

[8] D. Payton, M. Dally, R. Estkowski, M. Howard, and C. Lee. Pheromone robotics. *Autonomous Robots*, 11(3), 2001.

[9] *Khepera II*. 19 March 2007. http://www.k-team.com/kteam/index.php?page=17&rub=22&site=1

[10] *Flockbots*. 19 March 2007. http://cs1.gmu.edu/ eclab/ projects/robots/flockbots/